

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra měřicí a řídicí techniky

Využití LIN sběrnice v řídicích systémech
Utilization of the LIN Bus in Control Systems

2010

Vít Otevřel

Zadání bakalářské práce

Student: **Vít Otevřel**
Studijní program: B2649 Elektrotechnika
Studijní obor: 2601R004 Měřicí a řídicí technika
Téma: **Využití LIN sběrnice v řídicích systémech**
Utilization of the LIN Bus in Control Systems

Zásady pro vypracování:

1. Obecný popis komunikačního rozhraní LIN.
2. Vytvoření řídicího obvodu s mikrokontrolérem s LIN rozhraním a LCD displejem.
3. Realizace master/slave komunikace mezi dvěma jednotkami.
4. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

1. ŠVIMBERSKÝ, Z. LIN - Local Interconnect Network (Bakalářská práce, Praha, 2007)
2. Internetové stránky LIN konzorcia (<http://www.lin-subbus.de>)

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Zdeněk Slanina, Ph.D.**

Datum zadání: 20.11.2009

Datum odevzdání: 07.05.2010



doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry



prof. Ing. Ivo Vondrák, CSc.
děkan fakulty

Prohlášení studenta

„Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.“

V Ostravě dne

podpis

Abstrakt

Tématem této bakalářské práce je komunikace po sběrnici LIN (Local Interconnect Network). Jde o jednoduchou a levnou průmyslovou sběrnici, která se používá nejčastěji v automobilovém průmyslu. Cílem je seznámit se s touto sběrnici, navrhnout a naprogramovat obě jednotky (nadrženu i podrženu), které spolu budou komunikovat. Podřízená jednotka bude měřit teplotu, posílat ji řídicí jednotce a ta ji bude zobrazovat na displeji a posílat po sériovém rozhraní do PC.

Abstract

This bachelor work deals with communication over LIN (Local Interconnect Network) bus. It is simple and cheap industrial bus, which is most often used in automotive industry. The aim of this work is introduction of this bus and design and programming master and slave units, which will communicate together. Slave unit will measure temperature and send this value to master unit which will display this temperature and send to PC over serial interface.

Klíčová slova

Freescall, LIN, řídicí jednotka, MCU, Motorola, sběrnice, podržená jednotka.

Key words

Bus, Freescall, LIN, master unit, MCU, Motorola, slave unit.

Seznam použitých symbolů a zkratk

BDM	Rozhraní pro programování/ladění MCU (Background Debug Module)
CAN	Controller Area Network
COM	Sériové rozhraní (RS-232)
CSV	Soubor určený pro výměnu tabulkových dat (Comma-separated values)
DPS	Deska plošného spoje
GND	Zem (Ground)
ID	Identifikátor
IDE	Integrated Development Environment
LCD	Displej (Liquid Crystal Display)
LIN	Local Interconnect Network
LSB	Nejméně významný bit (Least Significant Bit)
MCU	Mikrokontrolér
MSB	Nejvýznamnější bit (Most Significant Bit)
OOP	Objektově orientované programování
PC	Počítač (Personal Computer)
PDF	Přenosný formát dokumentů (Portable Document Format)
SCI	Sériové komunikační rozhraní (Serial Communication Interface)
SLIC	Slave LIN Interface Controller
UART	Universal Asynchronous Receiver Transmitter
XOR	Nonekvivalence

Poděkování

Zde bych rád poděkoval všem, kteří mi pomohli při tvorbě práce, zvláště pak panu Ing. Zdeňkovi Slaninovi PhD. za vytrvalou koordinaci a pomoc, a rodině za podporu ve studiu.

Obsah

1 Úvod.....	8
2 LIN (Local Interconnect Network).....	9
2.1 Základní charakteristika.....	9
2.2 Fyzická vrstva.....	10
2.3 Linková vrstva.....	10
2.3.1 Synchronizační pauza.....	11
2.3.2 Synchronizační pole.....	11
2.3.3 Identifikátor.....	11
2.3.4 Datový rámec.....	12
2.4 Komunikace.....	12
2.4.1 Master - Slave.....	12
2.4.2 Slave - Master.....	13
2.4.3 Slave - Slave.....	13
2.5 Změna režimu činnosti – uspání.....	13
2.6 Změna režimu činnosti – probuzení.....	13
3 Výběr součástek a návrh jednotek.....	14
3.1 LIN vysílač/přijímač MC33399.....	14
3.1.1 Základní vlastnosti.....	14
3.2 Digitální teploměr DS18B20.....	16
3.2.1 Základní vlastnosti.....	16
3.2.2 Použité příkazy v programu.....	17
3.3 Mikrokontrolér MC9S08DZ60.....	18
3.3.1 Základní vlastnosti.....	18
3.3.2 Sériové komunikační rozhraní.....	19
3.4 Mikrokontrolér MC9S08SL16.....	19
3.4.1 Základní vlastnosti.....	20
3.4.2 Hardwarová podpora LIN sběrnice.....	20
3.5 Hardware – řídicí a podřízená jednotka.....	21
3.5.1 Řídicí jednotka.....	22
3.5.2 Popis zapojení řídicí jednotky.....	23
3.5.3 Podřízená jednotka.....	24
3.5.4 Popis zapojení podřízené jednotky.....	24
4 Programování jednotek.....	25
4.1 Device Initialization.....	26
4.2 Řídicí jednotka.....	27
4.3 Podřízená jednotka.....	30
4.4 Komunikace řídicí jednotky s PC.....	32
5 Závěr.....	36
6 Použitá literatura.....	37
7 Seznam příloh.....	38

1 Úvod

Cílem bakalářské práce je seznámení se s LIN sběrnici a její využití v řídicích systémech. Prvním úkolem bylo seznámení se s LIN sběrnici pomocí doporučené literatury. Zjistit jak LIN sběrnice funguje, tzn. jaké napěťové úrovně používá, jak se po sběrnici komunikuje apod..

Druhým bodem zadání bylo vytvořit řídicí obvod s mikrokontrolérem, který by obsahoval LCD, LIN rozhraní pro připojení podřazeného zařízení a sériové rozhraní pro připojení PC. Bylo potřeba navrhnout i podřazené zařízení, které by také obsahovalo mikrokontrolér, LIN rozhraní a teplotní čidlo DS18B20.

Krok třetí zahrnoval realizaci komunikace master/slave mezi dvěma jednotkami a posílání teploty řídicí jednotkou do PC. To vyžadovalo vytvoření softwaru (firmwaru) do mikrokontrolérů firmy Freescale, které byly použity jak v řídicí, tak podřazené jednotce a obslužný program který bude zobrazovat (vizualizovat) teplotu v PC.

2 LIN (Local Interconnect Network)

První specifikace LIN sběrnice 1.0 byla vytvořena v červenci roku 1999. U zrodu této sběrnice bylo LIN sdružení, které se skládalo ze sedmi firem, Audi, BMW, DaimlerChrysler, Motorola, Volcano Communications Technologies, Volkswagen a Volvo. LIN sběrnice byla poprvé nasazena v roce 2001. Po deseti letech od vydání první specifikace LIN sběrnice je již na světě specifikace 2.1, která byla vytvořena na podzim roku 2008. LIN sběrnice je navržena jako levná automobilová sběrnice, aby nahradila CAN sběrnici tam, kde není potřeba její rychlost a vysoká bezpečnost přenosu. Sběrnice LIN tedy není náhrada za dražší CAN, ale pouze ji doplňuje. Na jeden automobil se předpokládá použití 3-10 LIN jednotek. Aplikace ve kterých je možné LIN najít jsou např. řízení dveří – okna, zrcátka, zámky; střešní okna, sedačky, klimatizace, stěrače, senzory, spínače a panely spínačů. Pro konstrukci zařízení s LIN nejsou třeba žádné speciální řadiče, proto můžeme použít běžný jednočipový mikrokontrolér s obvodem UART/SCI. Podřízená jednotka si vystačí i bez UART obvodu. Protože podřízené jednotky nemusí obsahovat přesné krystalové oscilátory, vystačí si i s levnými RC oscilátory. [2][4]



2.1 Základní charakteristika

Sběrnice se skládá z jedné jednotky master a jedné nebo několika jednotek slave, kdy maximum připojených jednotek slave je 16. Jde o jednovodičovou sériovou asynchronní sběrnici, která využívá pro sériový přenos dat formát UART. Počet slave jednotek je omezen počtem identifikátorů a fyzikálními vlastnostmi sběrnice. [2]

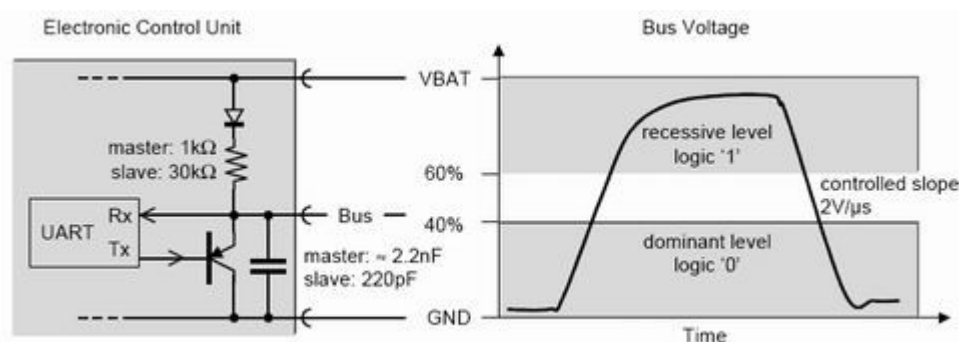
- Rychlost komunikace po sběrnici může být až 20 kb/s.
- Jednovodičová sběrnice s 12 V úrovní. To zajišťuje poměrně dobrou odolnost proti elektromagnetickému rušení.
- Komunikace po sběrnici je typu master-slave.
- Lze propojit až 17 jednotek (1x řídicí jednotka, 16x podřízených jednotek). Tento počet by neměl být překročen, každý uzel totiž snižuje impedanci o 3 % a při velkém počtu podřízených uzlů by byla impedance sítě již malá.
- Ve zprávě je možno přenést 0, 2, 4 nebo 8 bajtů.
- Maximální délka sběrnice by neměla přesáhnout 40 m.

- Zakončovací rezistor,
 - 1 k Ω pro řídicí jednotku,
 - 30 k Ω pro podřízenou jednotku.

2.2 Fyzická vrstva

Fyzická vrstva byla odvozena ze standardu ISO 9141. LIN sběrnice je realizována pouze jedním vodičem, to znamená, že je náchylnější na elektromagnetické rušení. Z tohoto důvodu musí být zajištěna malá rychlost přeběhu a tím bude pomalejší i přenos dat na sběrnici. Rychlost přenosu po sběrnici obvykle používá tyto rychlosti: 2400 (slow), 9600 (medium) a 19200 kb/s (fast).

LIN byla vyvinuta převážně pro automobilový průmysl, tomu jsou přizpůsobeny i napěťové úrovně. Logická 0 je reprezentována hodnotou napětí blízkému GND, tedy 0 V a logická 1 je reprezentována napětím blízkému VBAT, což odpovídá napětí kolem 12 V (Obr. 1). Spínače spojují sběrnici se zemí, stačí aby byl sepnutý jeden z nich a sběrnice přejde do stavu dominant (log. 0). Sběrnici zakončuje zdvihací rezistor, v případě jednotky master je to hodnota 1 k Ω a u jednotky slave 30 k Ω . Tyto zdvihací rezistory udržují stav recessive (log. 1) na sběrnici, když jsou spínače rozepnuté. Pro případ poruchy zdroje je zde v sérii s rezistorem zapojena dioda. Aby bylo zařízení odolnější proti elektromagnetickému rušení, připojují se paralelně k pinům LIN budiče kondenzátory. [2][4]

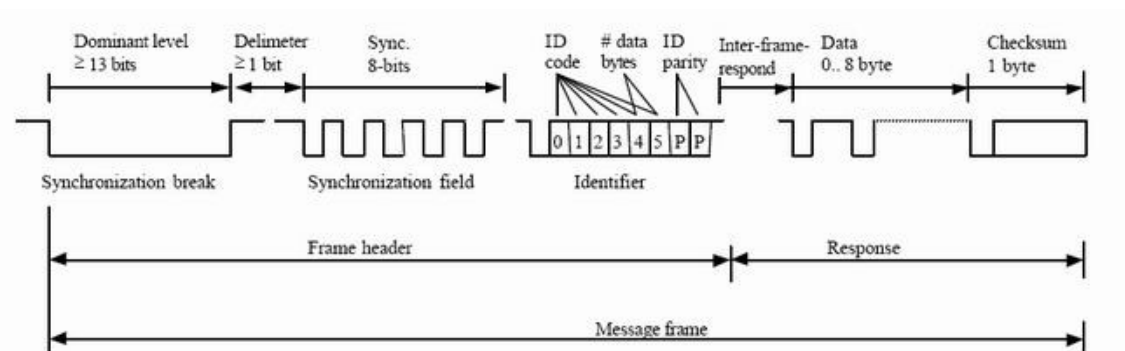


Obr. 2: Fyzická vrstva LIN sběrnice a napěťové úrovně [2]

2.3 Linková vrstva

LIN protokol je založený na UART protokolu, tzn. zprávy jsou kódovány po bajtech. Hlavní odlišností oproti UART-u je synchronizační pauza. Pomocí LIN zprávy, kterou vyšle nadřazená jednotka, se synchronizují, adresují podřízené jednotky a přenášejí data mezi zařízeními. Celý přenos po sběrnici je řízen řídicí jednotkou. LIN zpráva (LIN Message Frame) je složena ze dvou menších rámců, které odděluje mezi-rámcová mezera s variabilní délkou. [2]

- Hlavička (Header frame nebo Command frame) – vysílá jen master,
- odpověď (Response frame nebo Data frame) – vysílá master nebo slave.



Obr. 3: Formát rámce zprávy [3]

Hlavička se dále dělí na 3 části, synchronizační pauzu (Synchronization break), synchronizační pole (Synchronization field) a identifikátor (Identifier).

2.3.1 Synchronizační pauza

Synchronizační pauza je dlouhá minimálně 13 nulových bitů, aby byly schopny podřízené jednotky detekovat zprávy na sběrnici. [2]

2.3.2 Synchronizační pole

Toto pole slouží k synchronizaci hodin podřízených jednotek s řídicí jednotkou. Proto stačí když má přesné hodiny pouze nadřízená jednotka. Podřízené jednotky se synchronizují tak, že od sestupné hrany start bitu až po pátou sestupnou hranu synchronizačního bajtu změří čas a podělí ho 8, tím získají přenosovou rychlost řídicí jednotky. [2]

2.3.3 Identifikátor

Hlavička obsahuje i identifikační pole, které se ještě dělí na 6 bitový identifikátor a 2 bity zbudou na paritu (výpočet parity je na Obr. 4). Teoreticky tedy dostaneme až 64 identifikátorů. Čtvrtý a pátý bit ID pole určuje kolik dat se bude přenášet v datové části (0, 2, 4, nebo 8 bajtů). Uzly nemají fyzickou adresu, ale mají v paměti předdefinovaný seznam platných identifikátorů (Tab. 1). [2]

$$P0 = ID0 \oplus ID1 \oplus ID2 \oplus ID4$$

$$P1 = \neg (ID1 \oplus ID3 \oplus ID4 \oplus ID5)$$

Obr. 4: Výpočet parity [1]

Význam symbolů ve vzorci pro výpočet parity (Obr. 4):

- \oplus Nonekvivalence (XOR nebo Exclusive OR),
- \neg negace.

<i>Hodnota identifikátoru</i>		<i>Význam identifikátoru</i>
<i>dekadicky</i>	<i>hexadecimálně</i>	
0-59	0x00-0x3B	použito pro přenos obecných dat
60 a 61	0x3C a 0x3D	použito pro přenos diagnostických dat
62	0x3E	rezervováno pro uživatelem definovaný rámec
63	0x3F	rezervováno pro budoucí využití

Tab. 1: Přehled identifikátorů [4]

2.3.4 Datový rámec

Skládá se z 0 až 8 bajtů. Délka může být různá, záleží to na hardwarové a softwarové konfiguraci každého uzlu. Datový rámec taky obsahuje 1 bajt kontrolního součtu, který je počítán z přenášených dat. [4]

2.4 Komunikace

Řídící jednotka řídí veškerou komunikaci po sběrnici: [4]

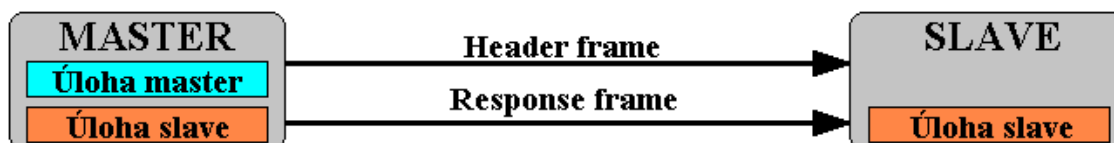
1. Definiuje přenosovou rychlost,
2. vysílá synchronizační pauzu, synchronizační pole a identifikátor,
3. monitoruje a potvrzuje data pomocí kontrolního součtu,
4. přepíná podřízených jednotek do spánku (sleep mód) a znovu je probouzí,
5. reaguje na signál probouzení od podřízených jednotek.

Podřízená jednotka má na starosti: [4]

1. Čekání na synchronizační impuls,
2. synchronizuje se podle synchronizačního pole,
3. dle identifikátoru provádí některou z činností:
 - nereaguje,
 - přijímá data,
 - vysílá data,
 - kontroluje nebo vysílá kontrolní součet.

2.4.1 Master - Slave

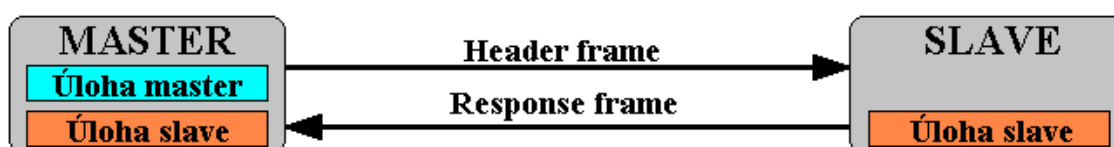
Nadřízená jednotka vyšle oba rámce (hlavičku i odpověď) jedné nebo více jednotkám podřízeným. Tento případ je typický pro ovládání akčních členů jednotky slave (zapínání/vypínání motorů, kontrolky, atd.).



Obr. 5: Komunikace master – slave [2]

2.4.2 Slave - Master

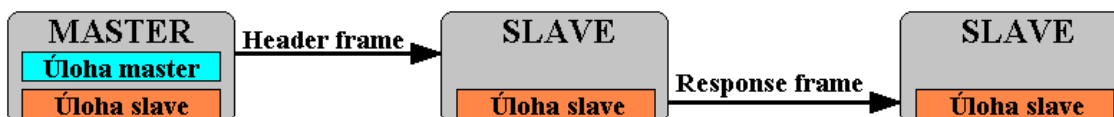
Master vyšle jen hlavičku a požaduje data od jednotky slave, která vyšle zpět jednotce master odpověď. Typický příklad pro čtení stavových informací z jednotky slave – stav vstupu (spínače), akčních členů, snímačů (senzorů), chyby atd.



Obr. 6: Komunikace slave - master [2]

2.4.3 Slave - Slave

Slave jednotka vysílá odpověď jedné nebo více slave jednotkám, bez průchodu dat jednotkou master. Používá se např. pro přímý přenos dat od smart senzorů k aktuátorům.



Obr. 7: Komunikace slave - slave [2]

2.5 Změna režimu činnosti – uspání

LIN specifikace podporuje uspání podřízené jednotky (jednotek). K tomu dochází zřídka kdy, např. když se po sběrnici komunikuje velmi málo a není nutná plná funkce jednotky slave. Nadřízená jednotka může uspání aktivovat datovým rámcem, který bude mít identifikátor 0x3C a první bajt dat bude roven 0x00 a vyšle ho všem jednotkám. Podřízené jednotky se poté přepnou do nízkopříkonového módu (sleep), dokud jednotka master nebo slave nevyšle budící (wake-up) signál. Pokud by např. vlivem rušení některý uzel nebo i všechny podřízené uzly nezachytily zprávu s povelům přechodu do uspávacího režimu, pak identifikují samy uspávací mód po určité době nečinnosti sběrnice $T_{TIME-OUT}$, která je většinou 4 s. [2]

2.6 Změna režimu činnosti – probuzení

Pokud je sběrnice LIN v nízkopříkonovém módu, je možné probudit sběrnici signálem určeným pro probuzení (Wake-up). Zaregistruje-li uzel slave nebo master podnět k probuzení a je-li nutné

předat tento nový stav dalším jednotkám, provede to vysláním signálu pro probuzení (Wake-up). Pokud není komunikační rychlost podřízeného uzlu synchronizována s uzlem nadřízeným, může být signál o 15 % delší nebo kratší, pak může nadřízená jednotka detekovat znaky 0xC0, 0x80, 0x00 (7 až 9 dominantních bitů). Po vyslání prvního probouzejícího signálu sleduje vysílající podřízená jednotka, zda dojde k zahájení vysílání synchronizační pauzy jednotkou řídící. Nestane-li se tak do doby T_{TOBRK} , pak podřízená jednotka opakuje vysílání signálu a to až třikrát. Pokud se ani takto nepovede probudit řídící jednotku, může podřízená jednotka vysílat další signál pro probuzení, ale až po uplynutí doby T_{T3BRK} , a trvá-li požadavek na probuzení. [2]

3 Výběr součástek a návrh jednotek

3.1 LIN vysílač/přijímač MC33399

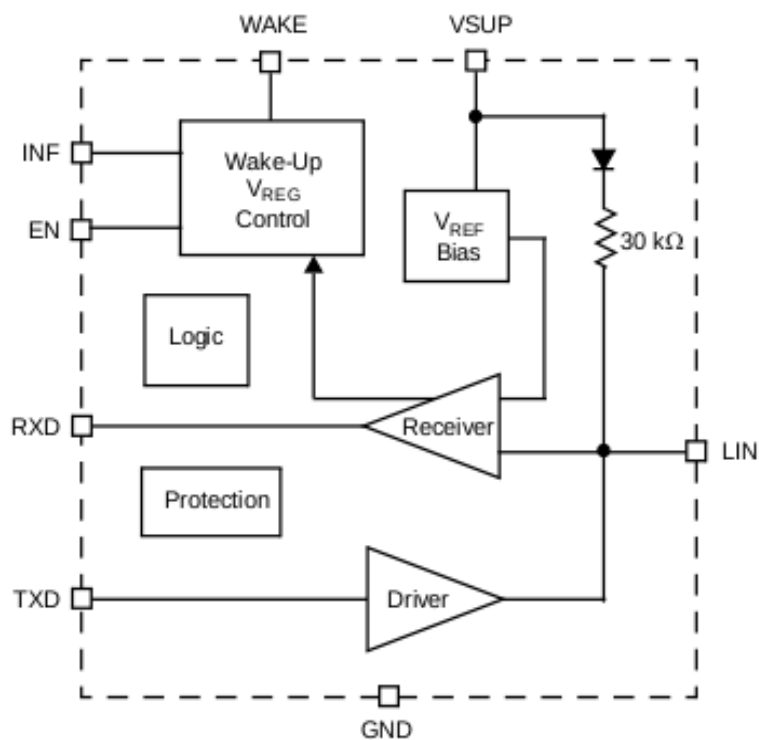
3.1.1 Základní vlastnosti

Obvod MC33399 je fyzické rozhraní mezi LIN sběrnici a mikrokontrolérem. Převádí napětíové úrovně LIN sběrnice na napětíové úrovně, které používají mikrokontroléry a většina logických obvodů (5 V logika). Po sběrnici může komunikovat rychlostí od 1 kb/s až do 20 kb/s. [7]

- Podporuje LIN protokol verze 1.3.
- Provozní napájecí napětí se pohybuje od 7 V do 18 V. Dokáže si poradit i s *napětíovou špičkou*, která může být velká až 40 V.
- 30 kΩ zdvihací rezistor.
- V nízopříkonovém módu (sleep mód) odebírá jen 20 μA.
- Řízení externího napětíového stabilizátoru.

Obvod je dodáván v SMD pouzdře SOIC 8. Má 8 vstupně/výstupních vývodů. Jednotlivé piny obvodu jsou vidět na Obr. 8. Na pin V_{SUP} (Power Supply) se přivádí napájecí napětí LIN sběrnice, které odpovídá asi 12 V (autobaterie). GND (Ground) pin je určen pro připojení země. Vývod LIN (LIN bus) je výstup na jeden datový vodič LIN sběrnice, po kterém se přijímají i vysílají data. U nadřízené jednotky musí být mezi pinem LIN (data) a V_{SUP} (napájení) zdvihací rezistor 1 kΩ v sérii s diodou. TxD (Data Input) je pin, který se připojí k mikrokontroléru (MCU obsahuje pin se stejným názvem TxD a ten patří periférii SCI, přes kterou je MCU připojen k LIN budiči, obdobné je to i s pinem RxD na MCU) a přes tento pin vysílá mikrokontrolér data na sběrnici. Naproti tomu RxD (Data Output) slouží k přijímání dat ze sběrnice, pin je samozřejmě taky připojen k mikrokontroléru. Pin EN (Enable Control) se používá pro určení módu (stavový diagram režimů je ukázán na Obr. 8) obvodu MC33399 (log. 1 znamená normální režim, log. 0 znamená režim spánku). INH (Inhibit Output) je určený pro řízení externího napětíového stabilizátoru, aktivuje jej nebo deaktivuje podle toho, zda je MC33399

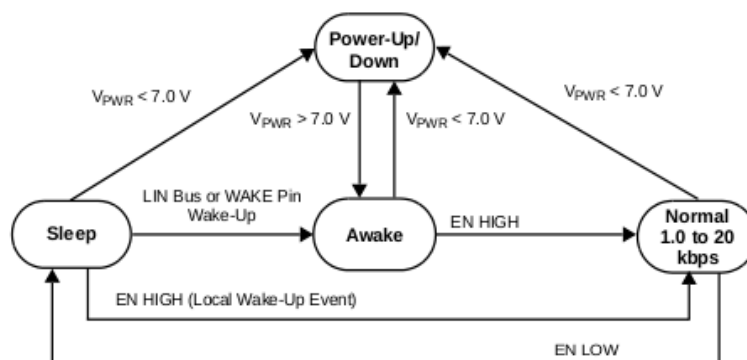
v normálním režimu nebo režimu spánku. Pin WAKE (Wake Input), slouží k probouzení z režimu spánku.



Obr. 8: Blokový diagram MC33399 [7]

Normální mód je režim, kdy obvod normálně vysílá a přijímá data. **Režim spánku** je režim, kdy je vysílací část obvodu vypnutá a obvod je v nízkopříkonovém režimu. V tomto režimu se spotřebováváný proud pohybuje kolem $20 \mu\text{A}$. Zařízení může být vzbuzeno z režimu spánku třemi způsoby (více na Obr. 9):

- Aktivitou na LIN sběrnici,
- interním probouzejícím uzlem na pinu EN (Wake-up),
- probuzení pomocí pinu WAKE.



Obr. 9: Stavový diagram režimů [7]

3.2 Digitální teploměr DS18B20

3.2.1 Základní vlastnosti

Digitální teploměr komunikuje s mikrokontrolérem po sběrnici 1-Wire. 1-Wire je jednovodičová sběrnice a proto pro komunikaci vyžaduje pouze jeden pin. Každé zařízení na této sběrnici má unikátní 64-bitový sériový kód uložený v ROM paměti, podle kterého se např. pozná, který teploměr vysílá teplotu je-li na sběrnici připojeno více teploměrů. Nevyžaduje žádné další externí součástky. Mezi další vlastnosti patří: [8]

- Napájecí napětí může mít rozsah od 3.0 V do 5.0 V.
- Měří teploty od -55 °C do 125 °C.
- V rozmezí od -10 °C do 85 °C má přesnost $\pm 0,5$ °C.
- Má uživatelsky nastavitelné rozlišení teploměru (9 až 12 bitů).
- Doba konverze teploty je max. 750 ms.
- Je možné použít funkci alarmu.
- Je dostupný ve třech pouzdrech:
 - 8-pin SO (150 mil),
 - 8-pin μ SOP,
 - 3-pin TO-92.

Uživatel si může vybrat jak přesně má DS18B20 teplotu měřit, to záleží na tom, jestli si vybere 9, 10, 11 nebo 12 bitové rozlišení (což odpovídá: 0,5 °C; 0,25 °C; 0,125 °C a 0,0625 °C). Standardně je nastaveno rozlišení 12 bitů. Teploměr vrací teplotu jako 16 bitové digitální slovo, tzn. vrací 2 bajty, LSB a MSB (Obr. 10). V závislosti na rozlišení trvá převod teploty na 2 bajty, při nejpřesnějším tedy 12-bitovém rozlišení trvá převod max. 750 ms. Teploměr obsahuje jeden konfigurační registr kde se nastaví rozlišení. Obsahuje i alarmy, tzn. má dva registry T_H a T_L , které porovnává s aktuálně naměřenou teplotou, je-li teplota menší než v registru T_L nebo naopak vyšší než v T_H , pak nastane alarm a nastaví se příslušný příznak. Mikrokontrolér může

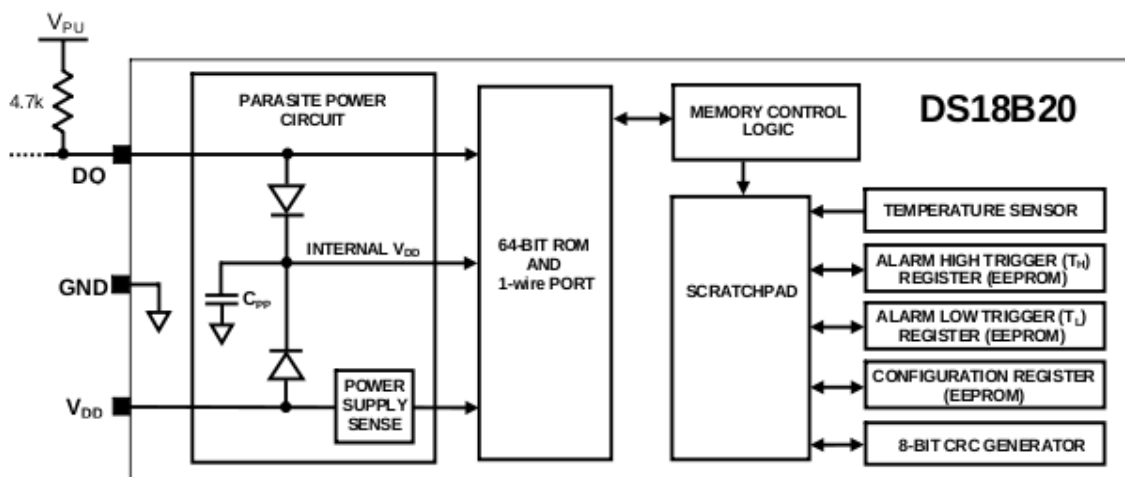
kontrolovat alarmy příkazem pro hledání alarmů (Alarm Search, 0xEC).

Pět pozic v MSB zabírají znaménkové bity, které určují zda je teplota kladná ($S = 0$) nebo záporná ($S = 1$).

LS Byte	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
MS Byte	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	S	S	S	S	S	2^6	2^5	2^4

Obr. 10: Formát teploty [8]

DS18B20 může být napájený buď externě nebo z datové linky, ovšem při napájení ze sběrnice musí být použit zdvihací MOSFET tranzistor.



Obr. 11: Blokové schéma DS18B20 [8]

R1	R0	Rozlišení	Max. doba převodu	
0	0	9-bitů	93,75 ms	$(t_{CONV}/8)$
0	1	10-bitů	187,5 ms	$(t_{CONV}/4)$
1	0	11-bitů	375 ms	$(t_{CONV}/2)$
1	1	12-bitů	750 ms	(t_{CONV})

Tab. 2: Možné rozlišení teploměru [8]

3.2.2 Použité příkazy v programu

Skip ROM – 0xCC

MCU může využít tento příkaz, pokud bychom chtěli poslat následující příkaz všem zařízením

na 1-Wire sběrnici, zařízení tedy nejsou za adresovány pomocí 64-bitového kódu. To může být užitečné pokud je vyžadováno např. aby všechny teploměry na sběrnici převedly teplotu najednou. Pokud bychom ale chtěli tento příkaz kombinovat s příkazem pro čtení teploty (0xBE), můžeme tak učinit jen v případě, že je na sběrnici připojen pouze jeden teploměr. [8]

Convert T – 0x44

Příkaz pro převod teploty. Následuje převod teploty a výsledky převodu uloží do 2-bytového registru. Když jsou výsledky převodu zapsány do paměti teploměru, vrátí se do nízkopříkonového stavu. [8]

Read Scratchpad – 0xBE

Příkaz dovolí MCU přečíst teplotu (již převedenou) z paměti teploměru. [8]

3.3 Mikrokontrolér MC9S08DZ60

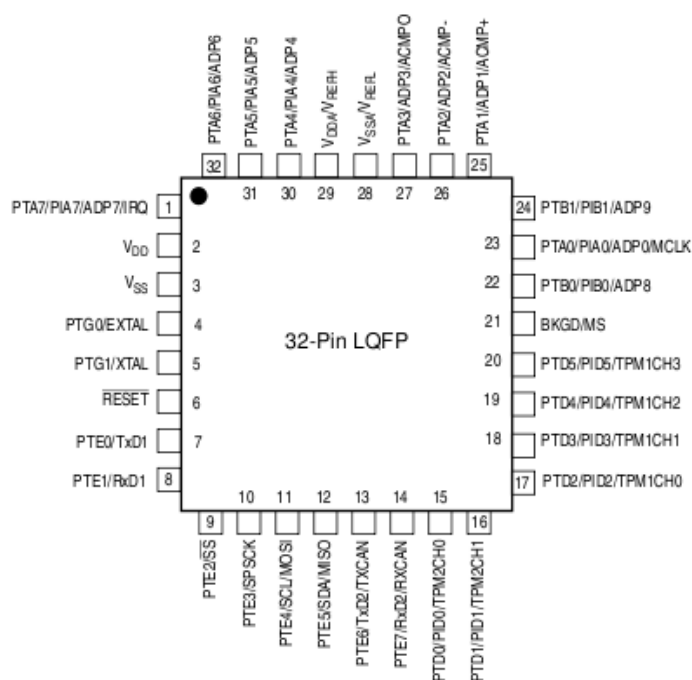
Použitý Mikrokontrolér v nadřazené jednotce je 8-bitový s jádrem HCS08. Obsahuje různé periferie a funkce, proto ho lze využít v mnoha oblastech. Hlavní oblast využití se ale předpokládá v automobilovém průmyslu, protože obsahuje rozhraní CAN a LIN (SCI/UART).

Hlavním důvodem proč byl tento mikrokontrolér vybrán do řídicí jednotky bylo to, že má 2 SCI rozhraní. Jedním SCI bude připojený k LIN sběrnici a pomocí druhého SCI bude možnost komunikace po RS-232 s PC. Charakteristickou vlastností je i CAN rozhraní, pomocí kterého lze mikrokontrolér připojit ke CAN sběrnici. Jedním z požadavků bylo aby na řídicí jednotce byl i displej typu LCD. K tomuto účelu plně postačí 32 pinů, proto bylo zvoleno pouzdro LQFP32 (Obr. 12). Přesné označení MCU s tímto pouzdem je MC9S08DZ60AMLC. Když jsou splněny všechny požadavky a jsou připojeny všechny periferie, zůstává k dispozici volný jeden port (8 vstupně/výstupních pinů).

3.3.1 Základní vlastnosti

- Napájení od 2.7 do 5.5 V.
- Jádro HCS08 může běžet až na 40 MHz a vnitřní sběrnice až na 20 MHz.
- 60 kB flash.
- 4 kB RAM.
- 2 kB EEPROM.
- Můžeme připojit externí oscilátor nebo použít interní zdroj hodin (Multi-purpose Clock Generator).
- Má 26 I/O pinů.
- Má jednovodičové rozhraní pro programování a ladění (BDM).
- **Systémová ochrana:**
 - Watchdog – Computer Operating Properly (COP).
 - Detekce nízkého napětí – Low Voltage Detect.

- Neplatná detekce adresy – Illegal address detection.
- Bloková ochrana Flash paměti – Flash block protect.
- **Obsahuje tyto periferie:**
 - 1x CAN, 1x I2C, 2x SCI a 1x SPI.
 - Verze CAN protokolu je 2.0 A, B.
 - SCI podporují LIN ve verzi 2.0. Rozšířená podpora generování a detekce pauzy.
 - Obsahuje jeden 4-kanálový, 16-bitový časovač a jeden 2-kanálový, 16-bitový časovač.
 - Obsahuje jeden 10-kanálový, 12-bitový A/D převodník.
 - Má dva analogové komparátory a RTC (Real-time counter).



Obr. 12: Pouzdro LQFP32 [5]

3.3.2 Sériové komunikační rozhraní

Jde o periferii, která zajišťuje sériovou komunikaci. SCI v tomto MCU má podporu protokolu LIN verze 2.0. Proto je SCI vybavený některými rozšířeními, jako je generování synchronizační pauzy, v případě použití MCU v podřízené jednotce, je tu možnost detekce synchronizační pauzy. Další vlastností je probuzení na aktivní hranu.

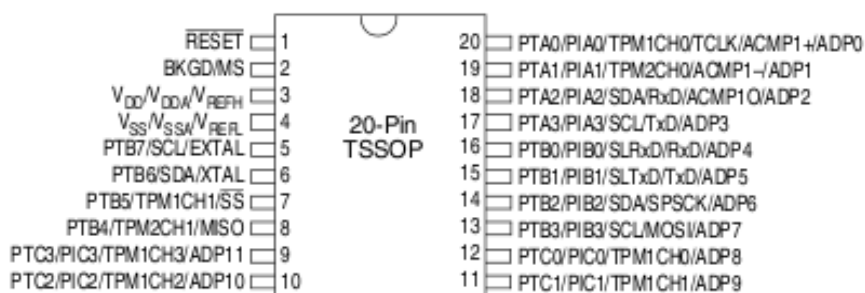
3.4 Mikrokontrolér MC9S08SL16

Mikrokontrolér který je použit ve podřízené jednotce je taky 8-bitový s jádrem HCS08. Tento MCU je přímo určený pro slave jednotky, má totiž periferii SLIC, což je LIN slave řadič, tedy

plně hardwarové řešení LIN-u. Protože podřízená jednotka bude měřit teplotu digitálním teploměrem DS18B20, který se připojuje jedním vodičem, bylo vybráno menší pouzdro. Vybraný MCU jehož přesné označení je MC9S08SL16CTJ je ve 20-pinovém pouzdře TSSOP20 (Obr. 13).

3.4.1 Základní vlastnosti

- Napájení od 2.7 do 5.5 V.
- Jádro HCS08 může běžet až na 40 MHz a vnitřní sběrnice až na 20 MHz.
- 16 kB flash.
- 512 B RAM.
- 256 B EEPROM.
- Možnost využití vnitřního (Internal Clock Source) nebo externího hodinového taktu.
- Má 16 I/O pinů.
- Má jednovodičové rozhraní pro programování a ladění (BDM).
- **Systémová ochrana:**
 - Watchdog – Computer Operating Properly (COP).
 - Detekce nízkého napětí – Low Voltage Detect.
 - Neplatná detekce adresy – Illegal address detection.
 - Bloková ochrana Flash a EEPROM paměti – Flash and EEPROM block protect.
- **Obsahuje tyto periférie:**
 - 1x SLIC, 1x I2C, 1x SCI a 1x SPI.
 - SLIC podporuje LIN ve verzi 2.0.
 - Obsahuje jeden 2-kanálový, 16-bitový časovač.
 - Obsahuje jeden 10-kanálový, 12-bitový A/D převodník.
 - Má dva analogové komparátory a RTC (Real-time counter).

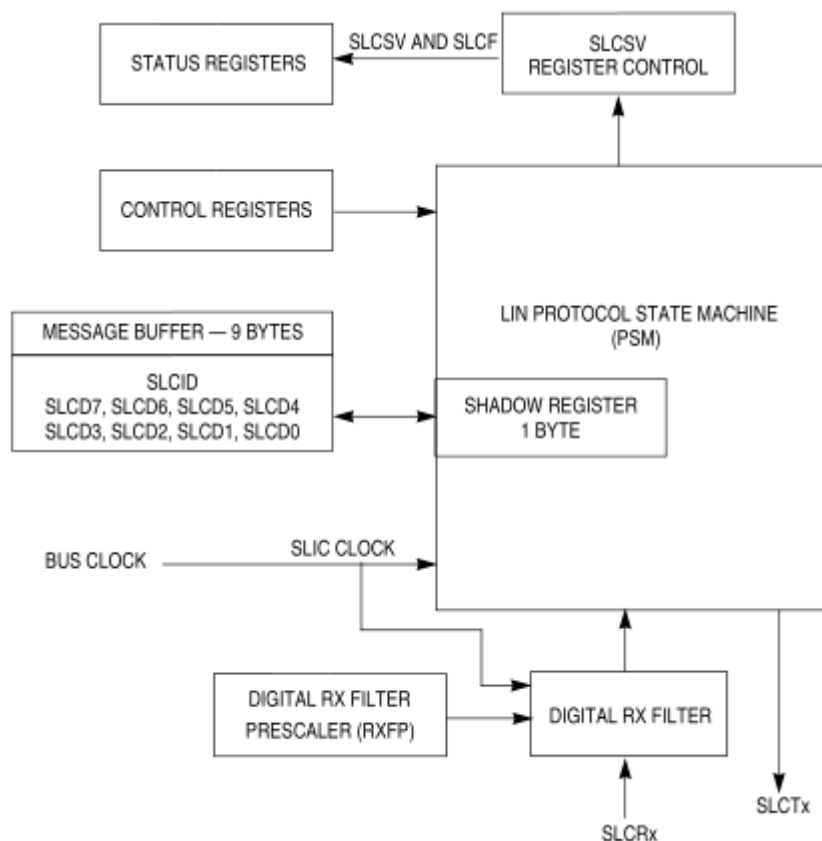


Obr. 13: Pouzdro TSSOP20 [6]

3.4.2 Hardwarová podpora LIN sběrnice

Tato periférie je použitelná jen pokud je MCU použitý v podřízené jednotce. Tento vestavěný SLIC modul velmi usnadňuje práci programátorům, protože SLIC zachycuje celou LIN zprávu

(identifikátor a 8-datových bytů), automaticky zpracuje a ověří synchronizační pauzu a synchronizační byte, automaticky nastaví rychlost podle synchronizačního pole. Automaticky generuje kontrolní součet a ověřuje ho. Podporuje protokol LIN ve verzi 2.0. Poskytuje plnou podporu ověřování chyb na LIN sběrnici.



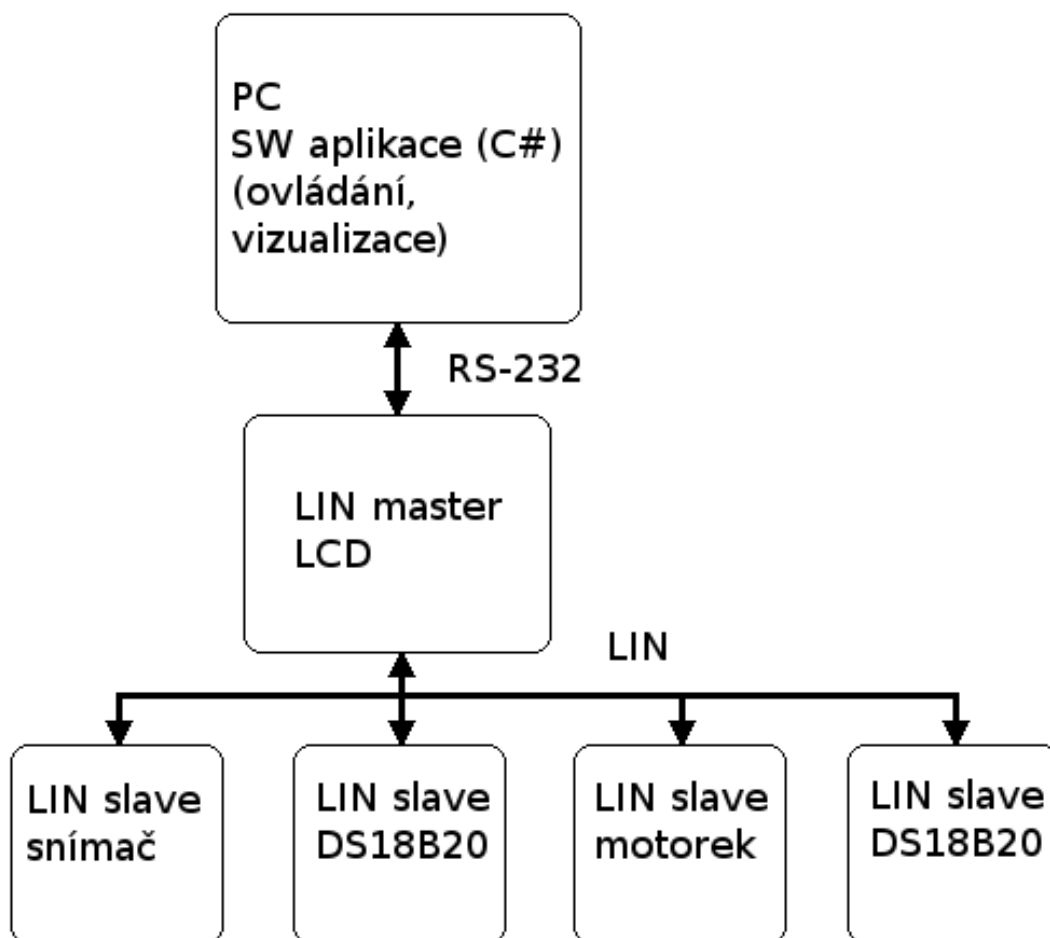
Obr. 14: Blokové schéma SLIC modulu [6]

3.5 Hardware – řídicí a podřízená jednotka

Zařízení master je spojeno se slave zařízením LIN sběrnici, po níž budou spolu komunikovat. Nadřazená jednotka má možnost připojení k PC přes sériový port. Na Obr. 15 je zobrazeno blokové schéma připojení zařízení.

LIN sběrnice je využitelná díky ceně nejen v automobilovém průmyslu, ale i tam, kde potřebujeme cokoli řídit nebo měřit. Konkrétně zde popisované zapojení lze použít tam, kde potřebujeme měřit teplotu, což bude prováděno pomocí podřízené jednotky a následně budou údaje přenesené po LIN-u do jednotky nadřazené a zobrazovány na LCD.

Propojení jednotky řídicí s PC nám dává další možnosti, teploty z řídicího zařízení mohou být zobrazovány na PC. Tím to ale nekončí, protože data, která jsou získána z řídicí jednotky mohou být vynesena do grafu, uložena do souboru a archivována.



Obr. 15: Blokové schéma propojení zařízení

3.5.1 Řídící jednotka

Zařízení master může být na LIN sběrnici jen jedno a to proto, že řídí komunikaci na LIN sběrnici. O řízení se stará 8-bitový mikrokontrolér firmy Freescale MC9S08DZ60 (je popsán výše), který obsahuje 2 SCI (Sériové komunikační rozhraní) rozhraní. Jedno SCI je použito pro připojení LIN sběrnice. Druhé SCI je potřeba pro sériovou komunikaci s PC po sériové lince (RS-232).

Důležité obvody jednotky master:

- MC9S08DZ60AMLC – mikrokontrolér jednotky master.
- MC33399 – LIN driver (budič).
- MAX232 – převodník úrovní RS-232 na TTL.
- LCD 2x16 znaků s řadičem.

3.5.2 Popis zapojení řídicí jednotky

Jak je uvedeno výše, LIN je 12 V sběrnice, protože byla primárně určena pro použití v automobilovém průmyslu. Mikrokontrolér, LCD a MAX232 potřebují ale pro své napájení 5 V, proto je na řídicí jednotce přítomen regulátor napětí 7805. Před regulátorem napětí je umístěna Shottkyho dioda, která plní funkci, jako ochrana před přepólováním. Obvod 7805, do něj vstupuje 12 V z LIN sběrnice, převádí těchto 12 V na 5 V pro napájení dalších obvodů na řídicí jednotce. Za stabilizátorem napětí 7805 je zelená LED dioda, která je určena pouze k signalizaci připojení jednotky k napájení. Před LED diodou je ochranným rezistor 1 k Ω .

Mikrokontrolér Freescale MC9S08DZ60AMLC je napájen 5 V ze stabilizátoru 7805. O rychlost (hodinový takt) MCU se stará krystal 16 MHz (v inicializaci MCU je takt snížený děličkou na 8 MHz), jenž je připojen na čtvrtý a pátý pin (piny EXTAL a XTAL) mikrokontroléru. Na pin 21 (BKGD) je připojen jediný vodič programovacího/ladícího konektoru. LIN driver je připojen k pinům periferie SCI1 v MCU a to dvěma vodiči, RxD a TxD. Ke druhé periférii SCI2 je připojen MAX232, opět dvěma vodiči, RxD a TxD. Celý PORT D je obsazen připojeným LCD. Protože zbyl volný celý PORT A, je vyveden na 10-pinový (2x5 pinů) konektor včetně napájení.

LIN budič MC33399 je taky od firmy Freescale. Obvod je napájený 12 V. Je-li budič použit v nadřazené jednotce musí mít mezi datovým pinem (LIN) a napájecím pinem (V_{SUP}) zapojený zdvihací rezistor 1 k Ω v sérii s diodou. Protože není použit externě regulovatelný napěťový regulátor, výstup budiče INH není připojen. Dále není využita možnost usnutí a probuzení, budič je tedy stále v normálním režimu, proto jsou vstupní piny ENABLE (EN) a WAKE-UP (WAKE) připojeny přímo k napěťovým úrovním které mají být na těchto pinech při normálním režimu: EN = log. 1, WAKE = log. 0. Nesmíme zapomenout na přijímací (RxD) a vysílací (TxD) piny budiče, které jsou taky připojeny k MCU. Pin RxD budiče je připojen k RxD pinu periferie SCI1 v MCU a pin TxD je zase připojen k pinu TxD periferie SCI1 v MCU.

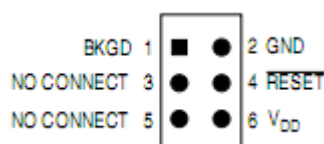
Řídicí jednotku je možno připojit k PC přes sériové rozhraní RS-232, proto je použit převodník s úrovní RS-232 (log. 0 je od +5 V (přijímač od +3 V) až +15 V (přijímač do +25 V a log. 1 je od -5 V (přijímač od -3 V) do -15 V (přijímač od -25 V), nedefinovaný stav je tedy od -3 V do +3 V) na TTL úrovní MAX232. Propojení je provedeno jednoduchým 3-drátovým propojením, kdy jsou společně spojeny piny 7 a 8 (piny CTS a RTS), piny 1, 4 a 6 (DCD, DTR a DSR) a nakonec i datové piny RxD a TxD. Ty jsou zapojeny takto: pin 2 na konektoru (RxD) je spojený s T1_{OUT} (TxD) pinem a pin 3 na konektoru (TxD) je spojený s R1_{OUT} (RxD) pinem na obvodu MAX232. Pro připojení obvodu MAX232 k MCU je použita periferie SCI2. Pin TxD2 v MCU je připojen k T1_{IN} pinu na budiči a obdobně je připojen i RxD2 pin (R1_{OUT} na budiči).

Pro zobrazování informací je použit LCD ATM1602B, který obsahuje řadič. Je to LCD s 2-mi řádky a 16-ti znaky na jeden řádek (2x16). Displej je připojen k PORTu D, který má přesně 6 pinů, což stačí pro připojení displeje. Podsvětlení displeje je připojeno permanentně k napájení (+5 V). Pin V0 je určen pro řízení jasu LCD a je připojen na potenciometr, kterým se dá řídit jas displeje. Komunikace s displejem využívá 4-bitovou komunikaci. Datové piny jsou

připojeny 4-mi vodiči. Proto musí být dolní polovina datových pinů (DB0 až DB3) displeje připojena k zemi, jelikož nejsou používány. Horní polovina pinů (DB4 až DB7) je připojena k pinům PORT-u D (PTD0 až PTD3) na MCU. Dva zbývající řídicí piny displeje jsou taky připojeny k MCU, pin ENABLE (E) je připojený k pinu PTD4 na MCU a pin Register Select (RS) je připojen k pinu PTD5 na MCU.

Na mikrokontroléru zůstal volný celý jeden port a to PORT A. Všech 8 pinů portu je vyvedeno na konektor 2x5 pinů. Za vhodné bylo považováno vyvést na konektor i napájení (+5 V a GND). Ke konektoru je možno připojit třeba ovládací tlačítko/tlačítka nebo nějaké signalizační LED diody, atd..

BDM konektor je velmi důležitý, je to rozhraní, které je používáno pro programování a ladění (debug) programu (firmwaru) do MCU. Zapojení BDM konektoru je znázorněno na Obr. 16. Z MCU je pro programování/ladění vyveden pouze jeden signál a to BKGD (pin 21).



Obr. 16: BDM rozhraní [5]

Schéma zapojení a návrh DPS jsou v příloze A.

3.5.3 Podřízená jednotka

Podřízených jednotek může být na LIN sběrnici až 16. Jednotka čeká na povely či příkazy od jednotky nadřazené, která řídí komunikaci po LIN-u. Pro podřízenou jednotku byl použit mikrokontrolér firmy Freescale MC9S08SL16 (popsán výše), jehož hlavní výhodou je, že obsahuje vestavěný modul SLIC. Jeden pin na MCU je použit pro připojení digitálního teploměru DS18B20 po 1-Wire sběrnici.

Důležité obvody jednotky slave:

- MC9S08SL16CTJ – mikrokontrolér jednotky slave.
- MC33399 – LIN driver (budič).
- DS18B20 – digitální teploměr.

3.5.4 Popis zapojení podřízené jednotky

Obdobně jako u řídicí jednotky je i u podřízené jednotky řešeno napájení. Vstup regulátoru napětí, opět to je obvod 7805, je připojen na 12 V z LIN sběrnice. Výstup obvodu 7805 je 5 V pro napájení MCU a digitálního teploměru DS18B20. Před regulátorem napětí je umístěna Shottkyho dioda, která chrání obvod před přepólováním. Zelená LED dioda signalizuje, zda je připojena jednotka k napájení. Sériově s LED diodou je zapojen ochranný rezistor 1 kΩ.

Mikrokontrolér Freescale MC9S08SL16CTJ je napájen 5 V z regulátoru 7805. O jeho rychlost (hodiny) MCU se stará vnitřní zdroj hodin o frekvenci 31,25 kHz (v inicializaci MCU je takt zvýšen násobičkou na 8 MHz), tím odpadá připojení vnějších hodin (krystalu). Plošný spoj je navržen i pro použití s externím zdrojem hodinového taktu, jen ve skutečnosti nebude osazen. I slave jednotku je potřeba nějakým způsobem naprogramovat případně ladit, tzn. i zde je přítomen konektor BDM. Jeden pin MCU je zabrán digitálním teploměrem.

Budič LIN sběrnice je také stejný, MC33399 od firmy Freescale. Napájen je přímo z LIN-u 12 V. Zde je změna oproti řídicí jednotce v tom, že podřízené jednotky nemají zdvihací rezistor a datový pin LIN je připojen přímo ke sběrnici. Piny RxD a TxD jsou připojeny k MCU na piny SLRxD a SLTXD, ne na periférii SCI, ale vestavěný řadič SLIC. Ostatní piny LIN budiče jsou připojeny stejně jako u nadřazeného zařízení, a to následovně: vstup EN = log. 1, vstup WAKE = log. 0 a výstup INH není připojen.

Na 17 pinu (PTA3) je připojena pouze jedním datovým vodičem sběrnice 1-Wire digitálního teploměru DS18B20 firmy Dallas. Teploměr je napájen 5 V z regulátoru 7805. DS18B20 vyžaduje zdvihací rezistor 4k7 mezi datový pin DQ a napájení 5 V, aby byla na 1-Wire sběrnici log. 1 při nečinnosti. Paralelně k pinům DS18B20 je vyveden konektor pro případné připojení více zařízení (teploměrů či jiných) 1-Wire zařízení.

I když má tento mikrokontrolér pouze 20 pinů, i zde zůstaly některé volné. Vyvedeny jsou jen 4 volné piny (PTB4, PTB5, PTC2 a PTC3) a to z toho důvodu, že pokud by volných pinů bylo vyvedeno více nebo všechny, byl by složitější návrh DPS. Piny jsou vyvedeny na malý konektor 2x3 včetně napájení (+5 V a GND).

BDM konektor je velmi důležitý, je to rozhraní, které je používáno pro programování a ladění (debug) programu (firmwaru) v MCU. Zapojení BDM konektoru je na Obr. 16 (u popisu jednotky master). Z MCU je pro programování/ladění vyveden pouze jeden signál a to BKGD (pin 2).

Schéma zapojení a návrh DPS jsou v příloze B.

4 Programování jednotek

Pro vývoj softwaru do MCU byl použit Freescale CodeWarrior Development Studio for Microcontrollers V6.3. Je to integrované vývojové prostředí (IDE) s grafickým uživatelským prostředím od firmy Metrowerks, dnes Freescale. CodeWarrior je určený pro vývoj aplikací pro MCU Freescale (Motorola).

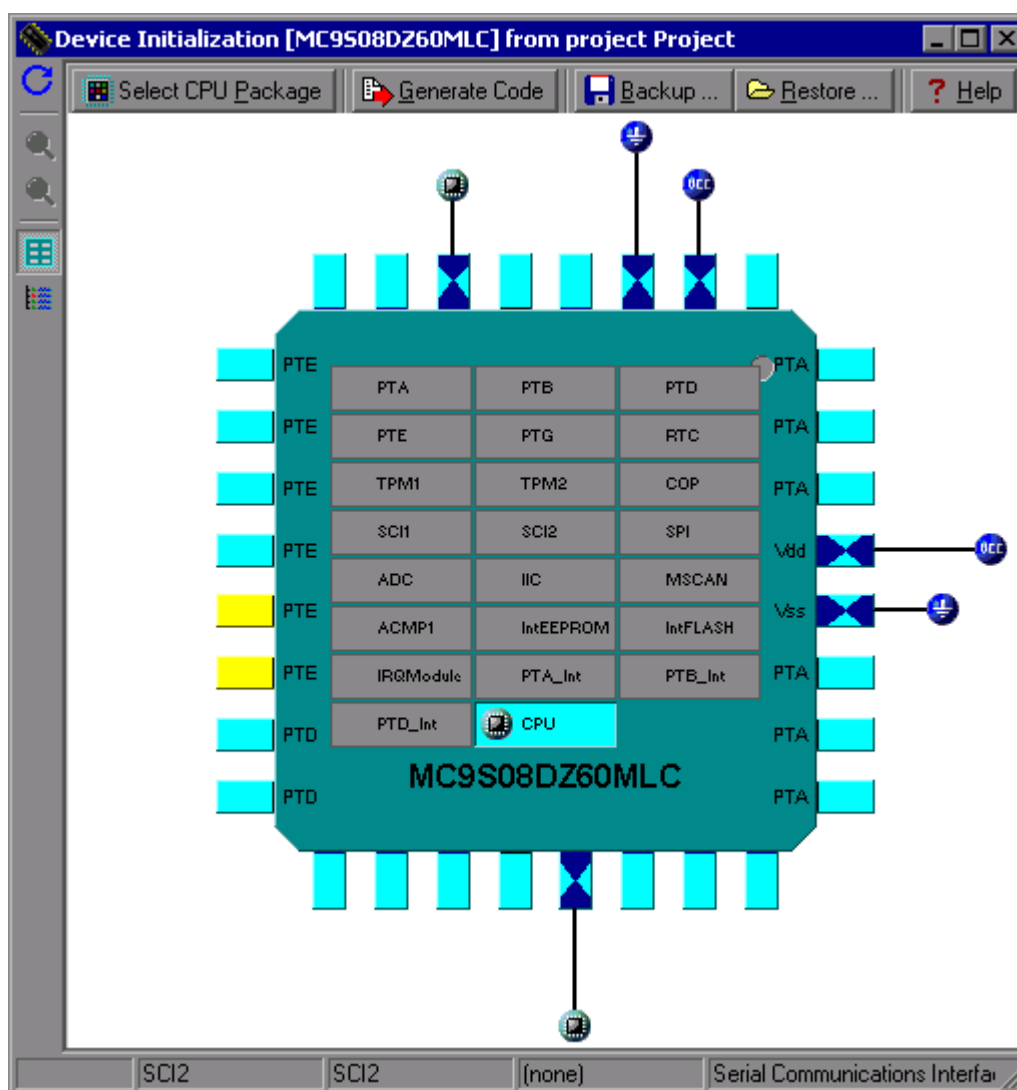
CodeWarrior obsahuje spoustu nástrojů, jako například správce projektů, C a C++ kompilátor a linker s možností optimalizace kódu, nástroj pro programátor/ladění, simulátor MCU a v neposlední řadě taky Processor Expert a Device Initialization (taky spadá pod Processor Expert). IDE obsahuje samozřejmě taky klasické funkce jako editor textu (zdrojových kódů), hledání a nahrazování výrazů v souboru, porovnávání souborů.

Processor Expert pomáhá vyvíjet vestavěné systémy, generuje ovladače, ale pouze v jazyku C, dovoluje hardwarově nezávislý přístup k perifériím MCU. Obsahuje knihovnu

komponent nazvanou Embedded Components. Embedded Components zapouzdří inicializaci a funkci základních prvků vestavěných systémů jako je jádro MCU, periférie MCU, samostatné periférie, virtuální zařízení a softwarové algoritmy. Tyto schopnosti jsou propojeny přes vlastnosti, metody a události (podobně jako objekty v OOP).

4.1 Device Initialization

Device Initialization poskytuje rychlou a jednoduchou cestu jak inicializovat MCU. Vygeneruje inicializační zdrojový kód pro vybraný MCU. Obsahuje jen jednu sadu komponent: Peripheral Initialization Components. Inicializační kód může být generován pro assembler nebo v jazyku C.



Obr. 17: Device Initialization

Při vytváření nového projektu je spuštěn průvodce, pomocí kterého zvolíme MCU (MC9S08DZ60 pro mastera a MC9S08SL16 pro slave) pro který budeme vyvíjet software a typ připojení MCU, respektive cílové DPS s MCU k PC (pro simulaci Full Chip Simulation a pro programování/ladění HCS08 Open Source BDM). V dalším kroku vybereme programovací jazyk ve kterém budeme vyvíjet, tedy jazyk C a zadáme umístění a jméno projektu. Dále chceme-li přidat do projektu nějaké existující soubory například driver pro LCD, hlavičkové soubory, najdeme je na disku a přidáme do projektu. Jako další krok vybereme Device Initialization, pomocí kterého vytvoříme inicializaci MCU. V předposledním kroku vybereme úroveň startovacího kódu (ANSI startup code), paměťový model (Small) a formát proměnných typu float a double (None) a dáme dokončit.

Po dokončení průvodce se otevře nové okno s inicializací MCU (Obr. 17). V okně uvidíme pouzdro MCU a uvnitř periferie které obsahuje. Změnu pouzdra MCU lze lehce změnit tlačítkem „Select CPU Package“. Kliknutím na periferii uvnitř pouzdra MCU se otevře další okno, ve kterém nastavíme vlastnosti dané periferie. Například pro periferii SCI2, se dá nastavit rychlost přenosu, formát dat, parita, zda budeme jenom přijímat nebo i vysílat data, přerušení a mnoho dalšího. Je-li vše nastaveno, zbývá vygenerovat inicializační kód kliknutím na tlačítko „Generate Code“. Ještě před vygenerováním kódu vybereme, zda má být inicializační kód v programovacím jazyku assembler nebo C a zda má kód přidat do našeho projektu. Vybereme jazyk C a zvolíme vytvořený inicializační kód do projektu.

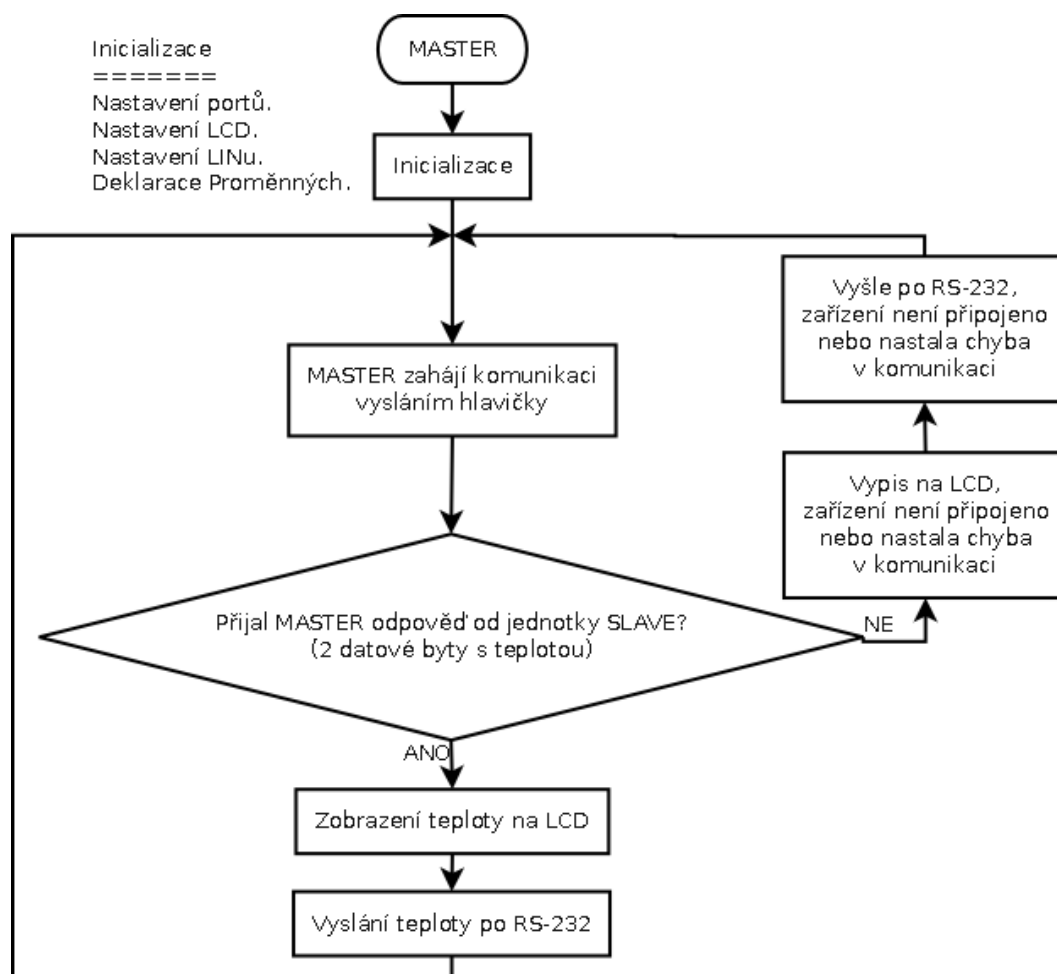
4.2 Řídicí jednotka

Řídicí jednotka pracuje podle vývojového diagramu na Obr. 18. Hlavní program je napsán v souboru *main.c*. První řádky souboru obsahují vložené hlavičkové soubory, následuje definice maker, deklarace globálních proměnných a deklarace funkcí. Inicializace, která je již napsaná v hlavní funkci *main()*, provádí veškerá počáteční nastavení. Tato inicializační funkce je vygenerována pomocí Device Initialization. Inicializace zahrnuje nastavení hodin MCU, nastavení používaných periférií (SCI1 pro LIN a SCI2 pro RS-232) a v neposlední řadě nastavení PORT-u D pro LCD.

Úspěšnou inicializací vstoupíme do hlavní (a nekonečné) smyčky programu *for(;;)*. V této smyčce jednotka vysílá hlavičku LIN zprávy a očekává odpověď od jednotky podřízené. Odpověď bude obsahovat 2 datové bajty s převedenou teplotou, kterou jednotka ji zobrazí na LCD a vyšle po sériovém portu. V případě že master vyslal hlavičku, ale nedostalo se mu odpovědi, měl by vypsát na displej chybové hlášení.

Hlavičkové soubory použité v programu:

- *delay.h*,
- *lcd.h*,
- *lin.h*,
- *main.h*.



Obr. 18: Vývojový diagram jednotky master

delay.h – obsahuje deklarace dvou funkcí, se zpožděním:

`void delay(unsigned int count);` – Zpoždění o maximální délce 210 ms, parametrem funkce je počet opakování cyklu `for`.

`void delay_1s(void);` – Zpoždění 1 sekunda.

lcd.h – obsahuje definici pinů MCU určených ke komunikaci s LCD a deklarace funkcí pro obsluhu LCD:

`void LCD_WriteData(unsigned char data);` – Funkce pro výpis jednoho znaku na LCD.

`void LCD_Init(void);` – Inicializace LCD.

`void LCD_Clear(void);` – Funkce vymaže LCD a nastaví pozici na počátek LCD.

`void LCD_Position(unsigned char radek, unsigned char sloupec);` – Funkce pro nastavení pozice na LCD.

void LCD_WriteString(char retezec);* – Funkce pro výpis řetězce na LCD.

void LCD_WriteDecByte(unsigned char data); – Funkce pro výpis dekadického čísla na LCD.

lin.h – obsahuje počet bajtů v datovém rámci, stavy na LIN sběrnici, strukturu pro zprávu (obsahuje ID a data), strukturu pro celý LIN rámec a deklaraci těchto funkcí:

void LINInit(void); – Nastaví strukturu pro příjem LIN rámce.

unsigned char LINSendMsg(unsigned char master, Bool poslat_data, struct zprava msg) – Jedna z nejdůležitějších funkcí, které obsahuje řídící jednotka je funkce pro vyslání zprávy. Pokud funkce vrátí „1“, byla zpráva úspěšně odeslána a pokud „0“, pak došlo k chybě. Prvním parametrem funkce je typ jednotky, zda je naše zařízení master („1“) nebo slave („0“) a podle toho bude/nebude vysílána hlavička. Druhým parametrem je určeno zda budou posílány i data („1“), nebo v případě řídící jednotky jen hlavička. Poslední parametr je struktura typu zprava (definována v hlavičkovém souboru *lin.h*), která obsahuje identifikátor a datové pole o max. 8-mi bajtech.

*unsigned char LINGetMsg(unsigned char ziskat_data, struct zprava *msg)* – Pomocí této funkce získáme přijatou zprávu. Bude-li vrácená hodnota „1“, byla odpověď přijata, jinak došlo k chybě. První parametr znamená, že budeme chtít získat data. Druhý parametr je ukazatel na strukturu typu zprava, která obsahuje identifikátor a datové pole o max. 8-mi bajtech.

enum stav_linu LINCheckState(void); – Funkce vrací stav sběrnice, tento datový typ je definován v hlavičkovém souboru *lin.h* a mohou nastat tyto stavy:

- IDLE – nečinný stav. Bude se vysílat synchronizační pauza.
- BREAK – stav po vyslání synchronizační pauzy.
- SYNCH – stav po vyslání synchronizačního pole.
- PROTECTED_IDENTIFIER – stav po vyslání ID s paritou.
- DATA_0 až DATA_7 – stav po vyslání/příjmu dat.
- CHECKSUM – stav po přijetí kontrolního součtu.

void SCI_RX(void); – Funkce je používána pro příjem jednotlivých bytů zprávy. Je vepsána do obsluhy přerušení příjmu periferie SCI1. Funkce plní vnitřní strukturu přijatými daty. Když je struktura naplněná a nastal stav CHECKSUM, můžeme použít funkci **LINGetMsg(...)** pro získání přijatých dat.

main.h – obsahuje nadefinované ID pro obě podřízené jednotky, nadefinovaný chybový bajt který pošle podřízená jednotka v případě, že není připojený teploměr na 1-Wire sběrnici, definovaný ruční a automatický režim a deklarace funkcí použitých v *main.c*.

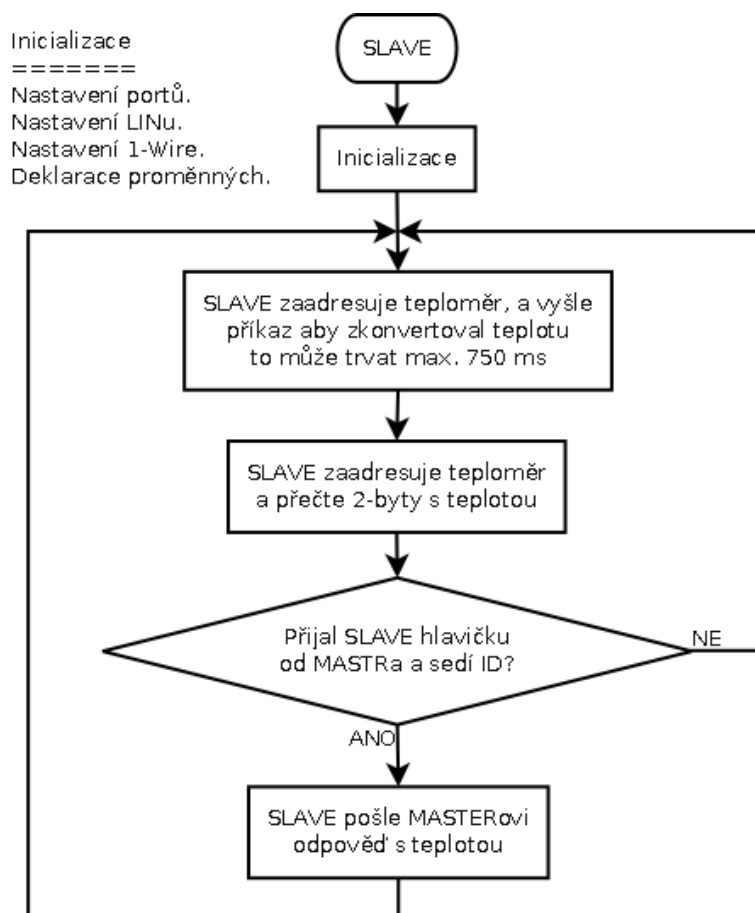
`void MasterWork(unsigned char ID, unsigned char LCD_row, unsigned char LCD_col);` – Ve funkci je zapouzdřeno vysílání hlavičky, přijetí dat od podřízené jednotky, zobrazení teploty na LCD a vysílání ID a teploty po RS-232. Funkce obsahuje 3 parametry, prvním parametrem je ID podřízené jednotky, které budeme posílat hlavičku respektive od které budeme zobrazovat teplotu. Další 2 parametry souvisí s pozicí na LCD, kam se má vypsát přijatá teplota.

`void ReceiveModeFromPC(void);` – Funkce přijímá z PC příkaz, zda má pracovat v automatickém nebo ručním režimu. Je vepsána do obsluhy přerušení příjmu periferie SCI2.

4.3 Podřízená jednotka

Na Obr. 19 je vývojový diagram jednotky slave. Inicializace a veškeré deklarace probíhají obdobně jako u jednotky master.

V hlavní cyklu `for(;;)` bude slave cyklicky měřit teplotu a ukládat do globální proměnné. Přijímání a vysílání dat a ID probíhá v přerušení SLIC modulu.



Obr. 19: Vývojový diagram jednotky slave

Hlavičkové soubory použité v programu:

- *delay.h*,
- *Iwire.h*,
- *slic.h*,
- *main.h*.

delay.h – obsahuje deklarace funkcí použitých v *delay.c* (obsahuje stejné funkce jako knihovna *delay* v jednotce master).

Iwire.h – obsahuje definici pinu MCU určený ke komunikaci s teploměrem a deklarace funkcí použitých:

unsigned char owReset(void); – Funkce pro reset 1-Wire sběrnice, kdy je očekáván pulz přítomnosti zařízení na sběrnici.

void owWriteByte(unsigned char data); – Funkce nejčastěji je používána pro vyslání příkazu teploměru.

unsigned char owReadByte(void); – Funkce používána pro přečtení teploty.

slic.h – obsahuje deklaraci funkce pro obsluhu přerušení SLIC modulu.

void SLIC_ISR(void); – Vše ohledně komunikace je prováděno v přerušení SLIC modulu, proto je nutné vložit volání této funkce do přerušení. Funkce pracuje tak, že z registru SLIC State Vector Register (SLCSV) zjistí kdo přerušení vyvolal, zda nastala chyba a jaká, bylo přijato ID, byly přijata data, atd., viz. Tab. 3. Na základě této proměnné se bude přepínat (větvit program pomocí *switch-case* příkazu) do jednotlivých stavů přerušení. Ke každému zdroji přerušení si pak můžeme napsat vlastní kód.

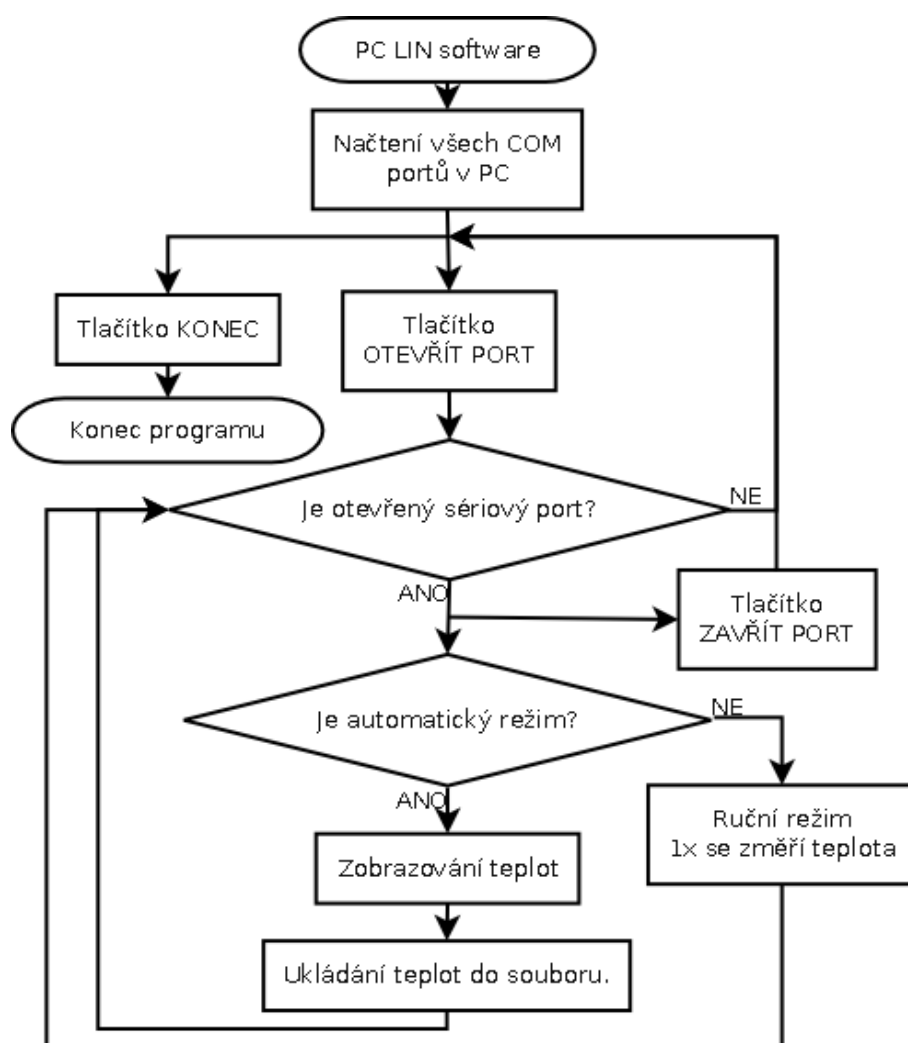
main.h – obsahuje, nadefinovaný chybový byte který pošle podřízená jednotka když není připojený teploměr na 1-Wire sběrnici.

SLCSV	I3	I2	I1	I0	Interrupt Source	Priority
0x00	0	0	0	0	No Interrupts Pending	0 (Lowest)
0x04	0	0	0	1	No-Bus-Activity	1
0x08	0	0	1	0	TX Message Buffer Empty Checksum Transmitted	2
0x0C	0	0	1	1	TX Message Buffer Empty	3
0x10	0	1	0	0	RX Message Buffer Full Checksum OK	4
0x14	0	1	0	1	RX Data Buffer Full No Errors	5
0x18	0	1	1	0	Bit-Error	6
0x1C	0	1	1	1	Receiver Buffer Overrun	7
0x20	1	0	0	0	Reserved	8
0x24	1	0	0	1	Checksum Error	9
0x28	1	0	1	0	Byte Framing Error	10
0x2C	1	0	1	1	Identifier Received Successfully	11
0x30	1	1	0	0	Identifier Parity Error	12
0x34	1	1	0	1	Reserved	13
0x38	1	1	1	0	Reserved	14
0x3C	1	1	1	1	Wakeup	15 (Highest)

Tab. 3: Možné zdroje přerušení SLIC modulu [6]

4.4 Komunikace řídicí jednotky s PC

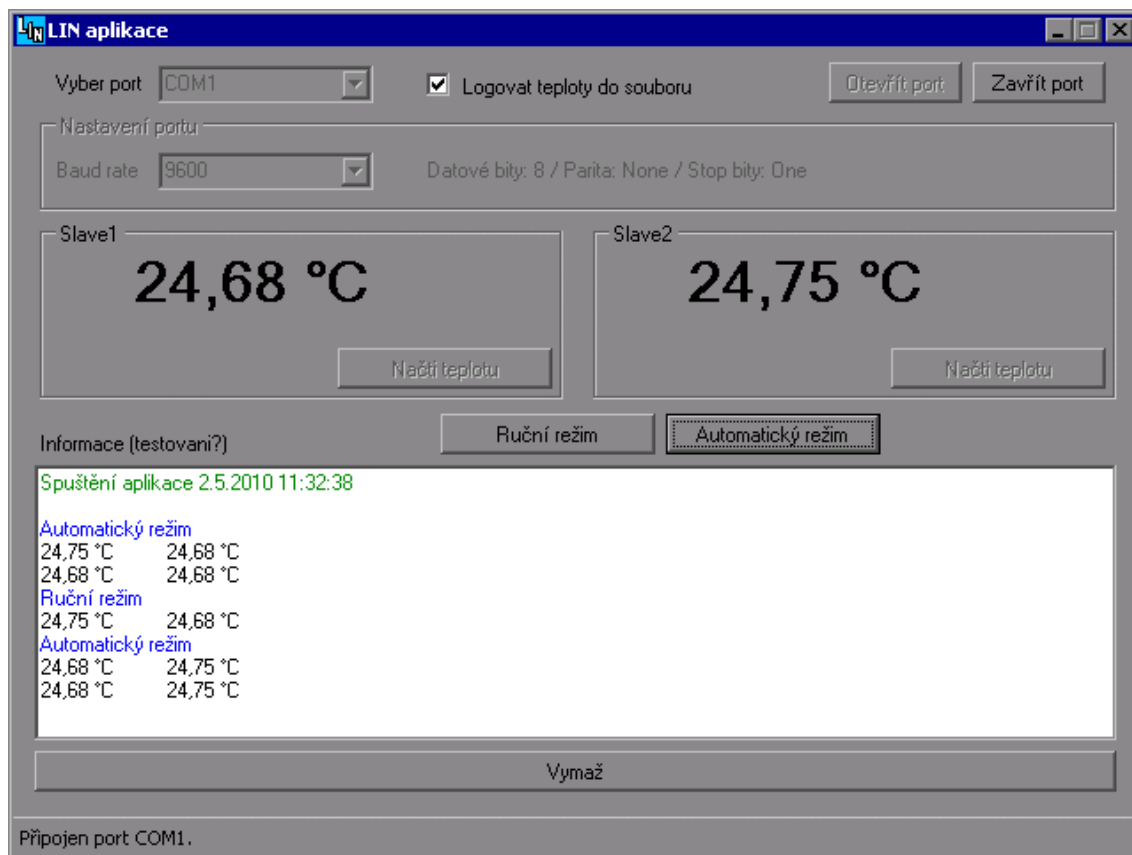
Chod programu je zobrazen na vývojovém diagramu na Obr. 20. Když je program spuštěn, jsou načteny všechny sériové porty v PC. Uživatel má možnost vybrat si s načtených COM portů ke kterému je LIN jednotka připojena. U nastavení portu lze nastavit jen rychlost, pevně je stanoven počet datových bitů na 8, parita žádná a stop bity 1. Uživatel si může vybrat, zda budou zobrazované teploty zároveň i ukládány do souboru. Tlačítko „Otevřít port“ otevře nastavený COM port a začne přijímat teplotu, oproti tomu tlačítko „Zavřít port“ ukončí přijímání teplot a port uzavře. Aplikace může běžet minimalizovaná v systémové oblasti, kde jsou v popisku zobrazovány teploty (Obr. 22).



Obr. 20: Vývojový diagram softwaru v PC

Aplikace byla vyvíjena v jazyku C#. Bylo použito vývojové prostředí Microsoft Visual C# 2008 Express Edition. Tuto verzi (Express Edition) IDE poskytuje Microsoft volně ke stažení na svých webových stránkách. Tato verze nesmí být používána pro výdělečnou činnost.

Díky uživatelskému prostředí, je grafický návrh aplikace velmi jednoduchý, z nástrojové sady (toolboxu) pouze přetahujeme objekty do okna tak, jak mají být rozmístěny. Dvojklikem na objekt se otevře zdrojový kód s jeho předvolenou událostí, kam můžeme psát vlastní kód.



Obr. 21: Aplikace pro zobrazování teploty v PC

Aplikace zobrazuje teploty tak jak je zobrazeno na Obr. 22. V systémové oblasti Windows jsou teploty zobrazeny následovně:



Obr. 22: Zobrazení v systémové oblasti

Se spuštěním aplikace se soubor s teplotami přepisuje, pro archivaci je potřeba soubor zkopírovat. Jednotlivé položky v souboru jsou odděleny středníkem. Každý řádek začíná datem, následuje čas kdy byla teplota změřena a konečně dvě teploty.

Formát dat uložených v souboru:

datum;	čas;	teplota1;	teplota2
28.4.2010;	18:21:26;	25,12;	24,87

Aplikace obsahuje také dva tlačítka na přepínání režimu – „Automatický režim“ a „Ruční režim“. Automatický režim funguje tak, že řídící jednotka cyklicky posílá žádost o teplotu jednotce podřízené a po přijetí odpovědi tuto teplotu zobrazí a pošle po COM-u.

V ručním režimu se změří teplota z obou podřízených jednotek pouze jednou a zobrazí na LCD a v PC aplikaci. Pokud bychom chtěli změřit teplotu znovu (jen jednou), je potřeba stisknout tlačítko „Ruční režim“.

5 Závěr

Cílem této bakalářské práce bylo seznámení se s LIN sběrnici, navrhnout hardware (jednotku master a slave) a realizovat komunikaci mezi těmito jednotkami. První část a to seznámení se sběrnici byla nutnost, bez toho bych nemohl navrhnout plošné spoje a realizovat komunikaci po této sběrnici. LIN sběrnice je popsána v druhé a zároveň nejrozsáhlejší kapitole.

Dalším krokem byl návrh desek plošných spojů jak jednotky řídicí tak podřízené. Proto bylo důležité se seznámit se všemi obvody, které byly na DPS osazeny. Jedním z důležitých obvodů pro komunikaci je budič LIN sběrnice, který převádí napěťové úrovně LIN sběrnice na úrovně, které používá SCI/UART. Neméně důležitý byl výběr správných mikrokontrolérů, aby měli všechny potřebné periferie respektive moduly. Oživování DPS probíhalo postupně, tzn. postupně jsem osazoval DPS a průběžně kontroloval funkčnost jednotlivých osazených obvodů.

Realizace komunikace po sběrnici mezi dvěma jednotkami byla dalším krokem. Aby mohly být zobrazovány teploty na řídicí jednotce, bylo nutno oživit displej a naučit se s ním pracovat. Měření teploty je obstaráno digitálním teploměrem DS18B20. Testování komunikace s teploměrem probíhalo na řídicí jednotce, kde byla komunikace vyladěna. Důležité pro přenos teplot z jednotky master do PC byla komunikace po sériovém komunikačním rozhraní. Práce s touto periferií byla odzkoušena na jednoduchém terminálu, kdy byly přes PC napsané znaky posílány po sériovém portu do řídicí jednotky a zobrazovány na LCD. Komunikace s podřízenou jednotkou byla odzkoušena vysíláním „1“ a „0“ jednotkou řídicí. Jednotka slave po přijetí „1“ rozsvítila LED diodu a naopak po přijetí „0“, byla LED dioda zhasnuta. Když bylo naprogramováno a otestováno odeslání odpovědi podřízené jednotky jednotce řídicí, byl do jednotky slave implementován kód pro měření teploty.

Vytvořil jsem aplikaci, která je určena pro PC s operačním systémem Windows. Je-li k sériovému portu PC připojena jednotka master, aplikace zobrazuje naměřené teploty. Je zde možnost ukládat naměřené hodnoty do souboru. Další naprogramovanou vlastností je možnost přepínání mezi automatickým a ručním režimem načítání teplot.

6 Použitá literatura

- [1] LIN konzorcium. *LIN Specification Package Revision 2.0* [online], 23.10.2003. <<http://www.lin-subbus.org/>>.
- [2] ŠVIMBERSKÝ, Z. *LIN – Local Interconnect Network*. Praha, 2007. 30 s. Bakalářská práce na Elektrotechnické fakultě Českého vysokého učení technického. Vedoucí bakalářské práce Ing. Jan Krákora.
- [3] RYLANDER, A. – WALLIN, E. *LIN – Local Interconnect Network – for use as sub-bus in Volvo trucks*. Göteborg, 2003, 73 s. Diplomová práce na Technické univerzitě v Chalmers. Vedoucí diplomové práce Björn Villing a Jan Söderberg.
- [4] SUTORÝ, Tomáš. *LIN (Local Interconnect Network)* [online]. C2004, poslední revize 10.3.2004. <<http://www.elektrorevue.cz/clanky/04012/index.html>>.
- [5] *MC9S08DZ60 Technical Data Sheet*. Rev. 4. Freescale Semiconductor, Inc., 2008. 416 s.
- [6] *MC9S08EL32: Data Sheet for MC9S08SL16 and MC9S08EL32 Series MCUs*. Rev. 3, Freescale Semiconductor, Inc., 2008. 356 s.
- [7] *MC33399, Local Interconnect Network (LIN) Physical Interface*. Rev. 8, Freescale Semiconductor, Inc., 2006. 19 s.
- [8] *DS18B20 Programmable Resolution 1-Wire Digital Thermometer*. Dallas Semiconductor, 2007. 21 s.
- [9] Příklad zdrojového kódu pro LIN master a slave jednotku [online], 2009. <<http://en.pudn.com/>>.
- [10] *AN2633/D: LIN Drivers for SLIC Module on the MC68HC908QL4*. Rev. 1. Freescale Semiconductor, Inc., 2004. 36 s.

7 Seznam příloh

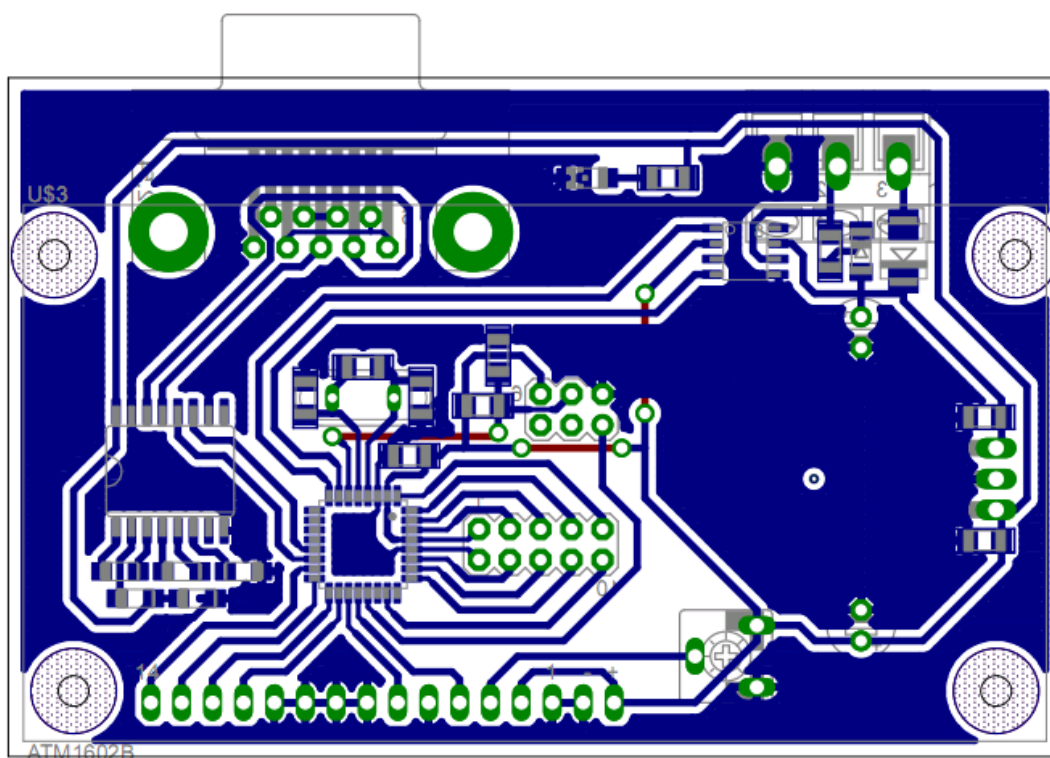
A Příloha.....	I
B Příloha.....	V
C Příloha.....	IX
D Příloha.....	IX

A Příloha

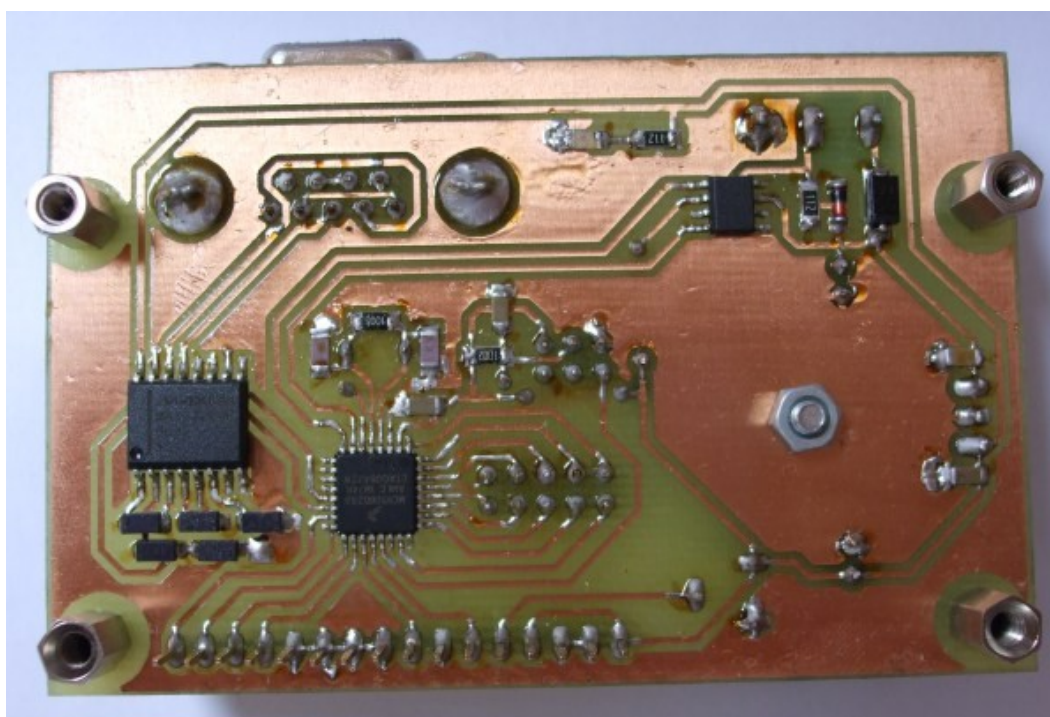
<i>Označení</i>	<i>(ks)</i>	<i>Hodnota</i>	<i>Pouzdro</i>	<i>Popis</i>	<i>Cena</i>
C1	1	330 nF	C1206	Keramický kondenzátor	3,00 Kč
C2, C6 a C7	3	100 nF	C1206	Keramický kondenzátor	6,00 Kč
C3 a C4	2	22 pF	C1206	Keramický kondenzátor	5,00 Kč
C5 a C13	2	10 uF	E2,5-6	Elektrolytický kondenzátor	3,00 Kč
C8 až C12	5	1 uF	SMCA	Tantalový kondenzátor	12,50 Kč
D1	1	10BQ060 (nebo SS16)	SMB	Shottkyho dioda, 60 V, 1 A	8,00 Kč
D2	1	1N4148	MINIELF	Dioda	1,00 Kč
IC1	1	7805	TO-220	Stabilizátor 5 V	7,50 Kč
IC2	1	MAX232CWE	SO16L	Převodník úrovní RS-232/TTL	25,00 Kč
LED1	1	zelená	1206	LED, zelená	3,00 Kč
Q1	1	QM 16MHZ	QS	Krystal mini, 16 MHz	8,00 Kč
R1 a R4	2	1 kΩ	R1206	Rezistor	4,00 Kč
R2	1	10 kΩ	R1206	Rezistor	2,00 Kč
R3	1	10 MΩ	R1206	Rezistor	2,00 Kč
R5	1	CA6VK025	CA6V	Trimr ležatý, 25 kΩ	5,00 Kč
SV1 a SV2	1	S2G20	----	Dvouřadá lámací lišta, 20 pinů	4,00 Kč
U\$1	1	MC9S08DZ60 AMLC	LQFP32	Mikrokontrolér	191,50 Kč
U\$2	1	MC33399	SOIC 8	LIN driver – fyzická vrstva	14,50 Kč
U\$3	1	ATM1602B	----	LCD	125,00 Kč
X1	1	AKZ 300/3	AKZ 300/3	Svorkovnice, 3 piny, rozteč 5.08	5,00 Kč
X2	1	CAN 9 Z 90	DSUB	Konektor CANON 9 pinů	9,00 Kč
----	1	DO1A	----	Chladič	3,00 Kč
----	1	S1G20	----	Jednořadá lámací lišta, 20 pinů	4,00 Kč
----	1	BL20G	----	Dutinková lišta, 20 pinů	10,00 Kč
CENA CELKEM					461,00 Kč

Tab. A.1: Seznam součástek potřebných pro jednotku master

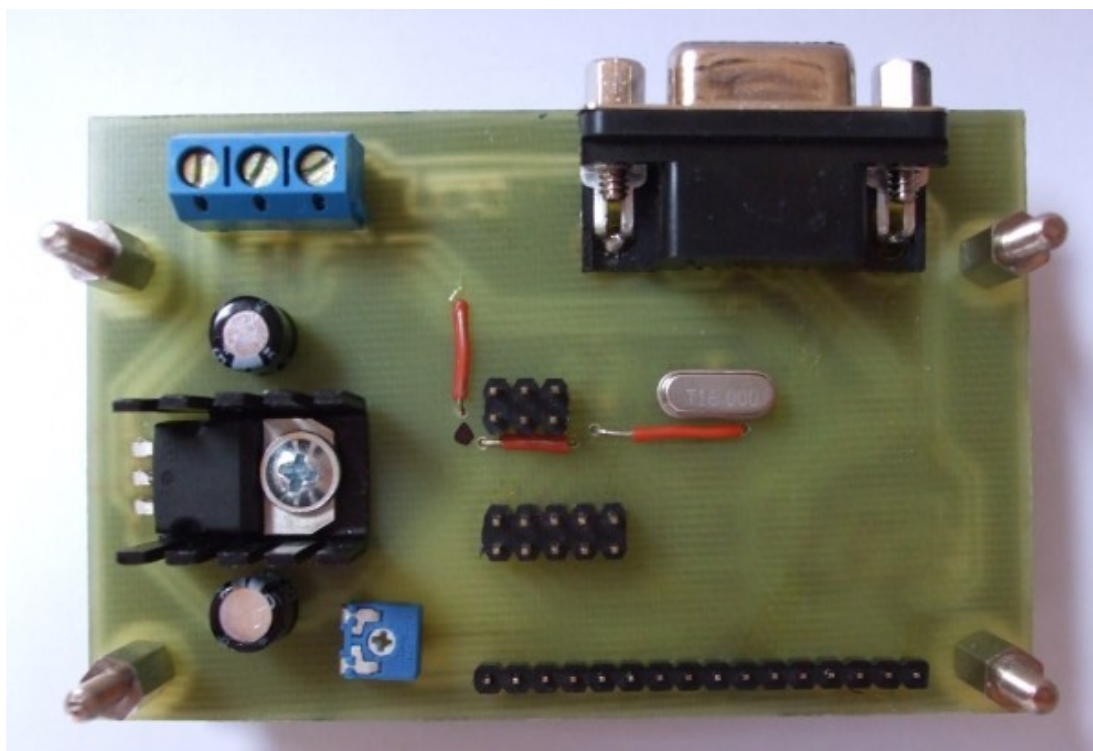
Uvedené ceny jsou z [GM Electronic](#), jen mikrokontrolér a LIN budič, jsou z obchodu [Farnell](#).



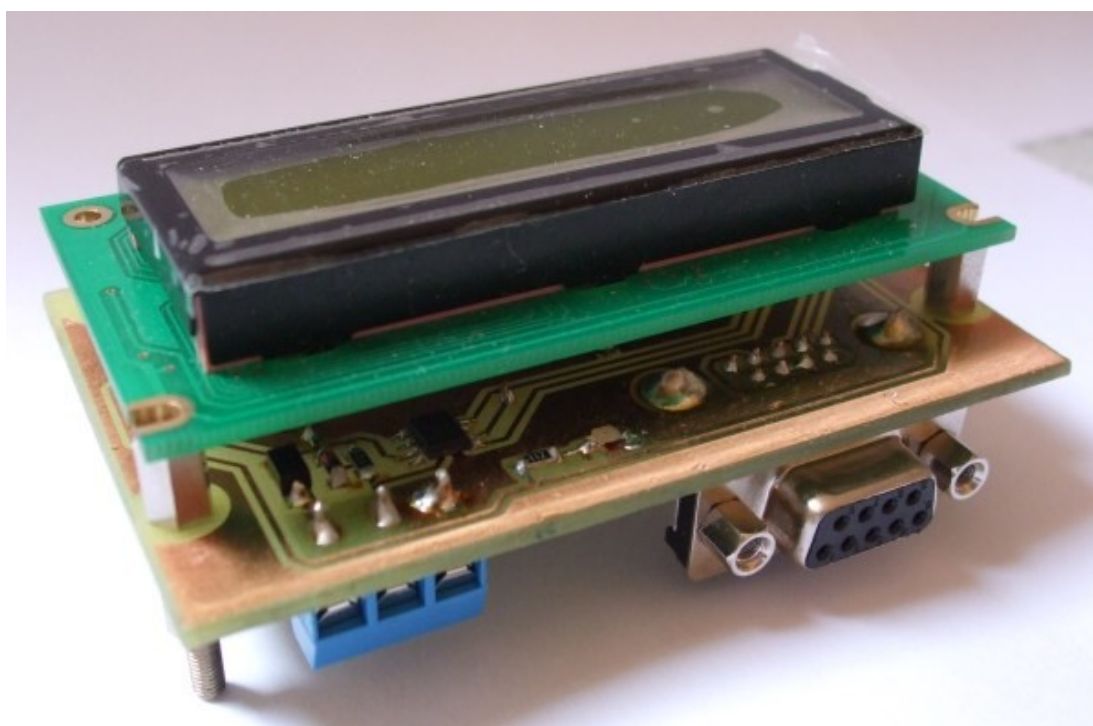
Obr. A.2: Návrh DPS jednotky master, pohled ze strany mědi (SMD součástek), rozměry 86x55 mm



Obr. A.3: Pohled na DPS jednotky master ze strany SMD



Obr. A.4: Pohled na DPS jednotky master ze strany součástek



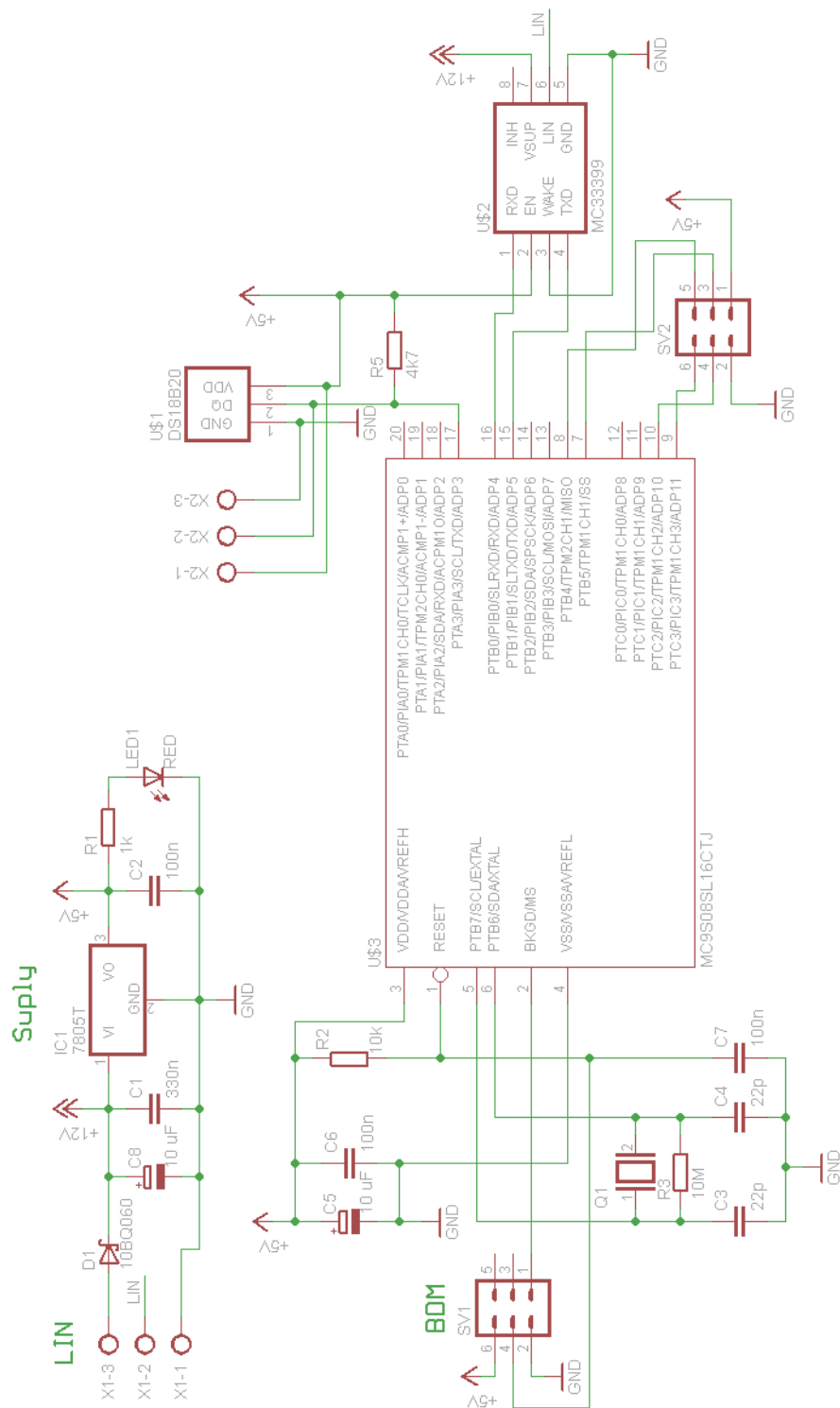
Obr. A.5: Celkový pohled na jednotku master

B Příloha

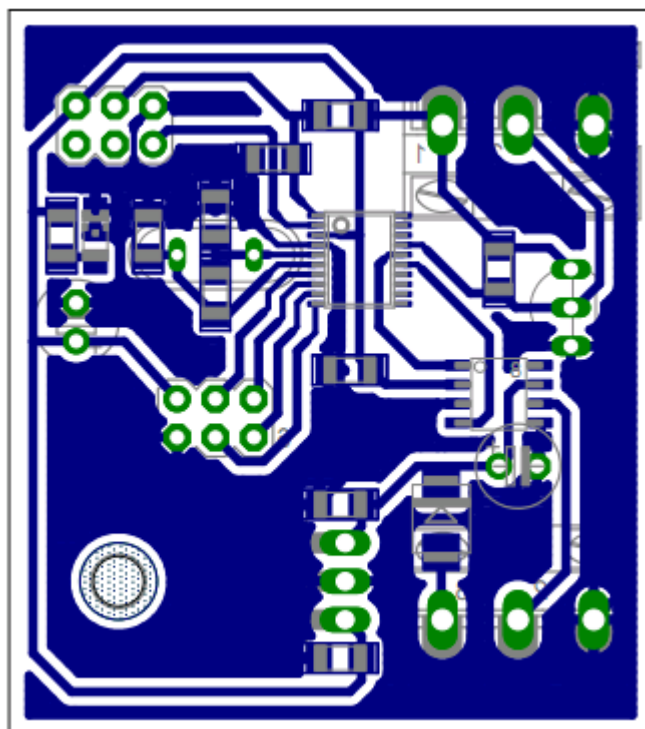
<i>Označení</i>	<i>(ks)</i>	<i>Hodnota</i>	<i>Pouzdro</i>	<i>Popis</i>	<i>Cena</i>
C1	1	330 nF	C1206	Keramický kondenzátor	3,00 Kč
C2, C6 a C7	3	100 nF	C1206	Keramický kondenzátor	6,00 Kč
C5 a C8	2	10 uF	E2,5-6	Elektrolytický kondenzátor	3,00 Kč
D1	1	10BQ060 (nebo SS16)	SMB	Shottkyho dioda, 60 V, 1 A	8,00 Kč
IC1	1	7805	TO-220	Stabilizátor 5 V	7,50 Kč
LED1	1	zelená	1206	LED, zelená	3,00 Kč
R1	1	1 kΩ	R1206	Rezistor	2,00 Kč
R2	1	10 kΩ	R1206	Rezistor	2,00 Kč
R5	1	4,7 kΩ	R1206	Rezistor	2,00 Kč
SV1 a SV2	1	S2G20	----	Dvouřadá lámací lišta, 20 pinů	4,00 Kč
U\$1	1	DS18B20	TO-92	Digitální teploměr	50,00 Kč
U\$2	1	MC33399	SOIC 8	LIN driver – fyzická vrstva	14,50 Kč
U\$3	1	MC9S08SL16 CTJ	TSSOP20	Mikrokontrolér	86,50 Kč
X1 a X2	2	AKZ 300/3	AKZ 300/3	Svorkovnice, 3 piny, rozteč 5.08	10,00 Kč
----	1	DO1A	----	Chladič	3,00 Kč
CENA CELKEM					204,50 Kč

Tab. B.1: Seznam součástek potřebných pro jednotku slave

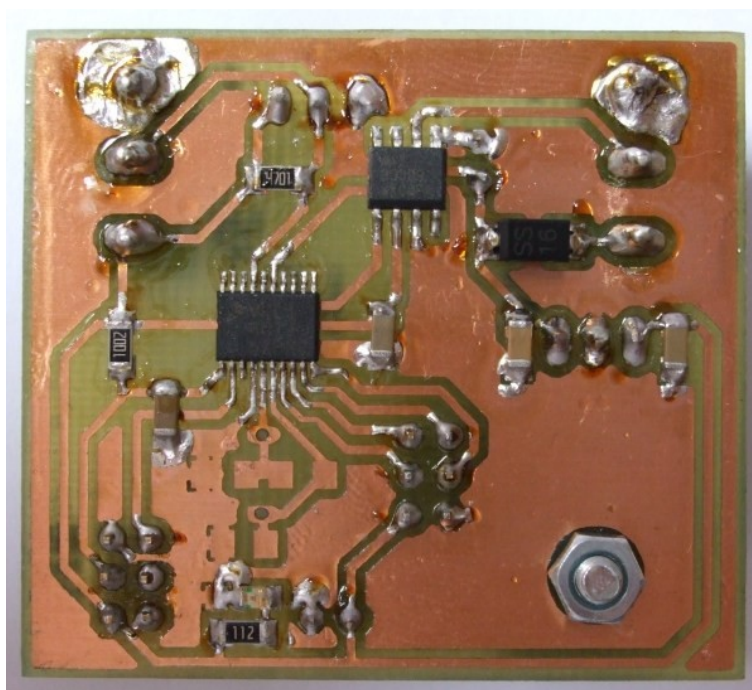
Uvedené ceny jsou z [GM Electronic](#) jen mikrokontrolér a LIN budič, jsou z obchodu [Farnell](#).



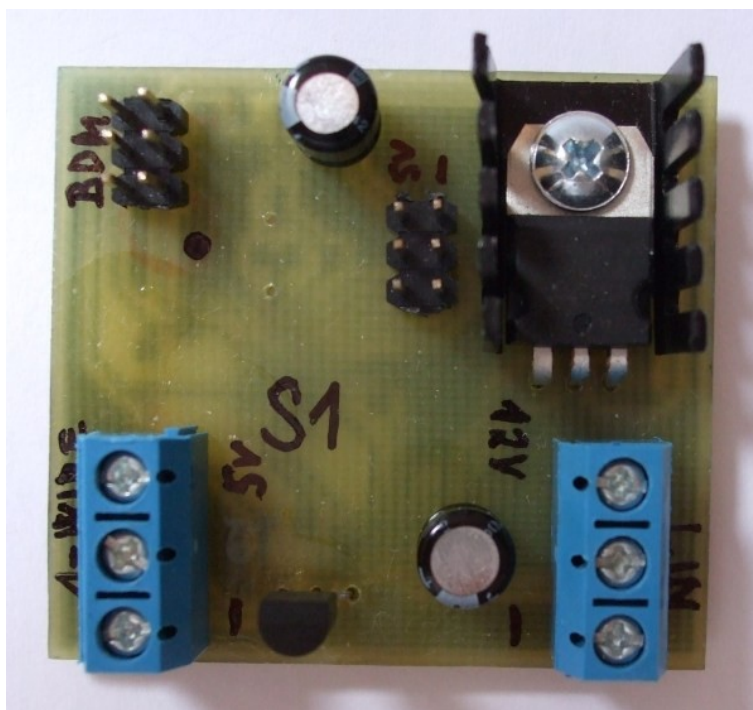
Obr. B.1: Schéma jednotky slave



Obr. B.2: Návrh DPS jednotky slave, pohled ze strany mědi (SMD součástek), rozměry 42x47 mm

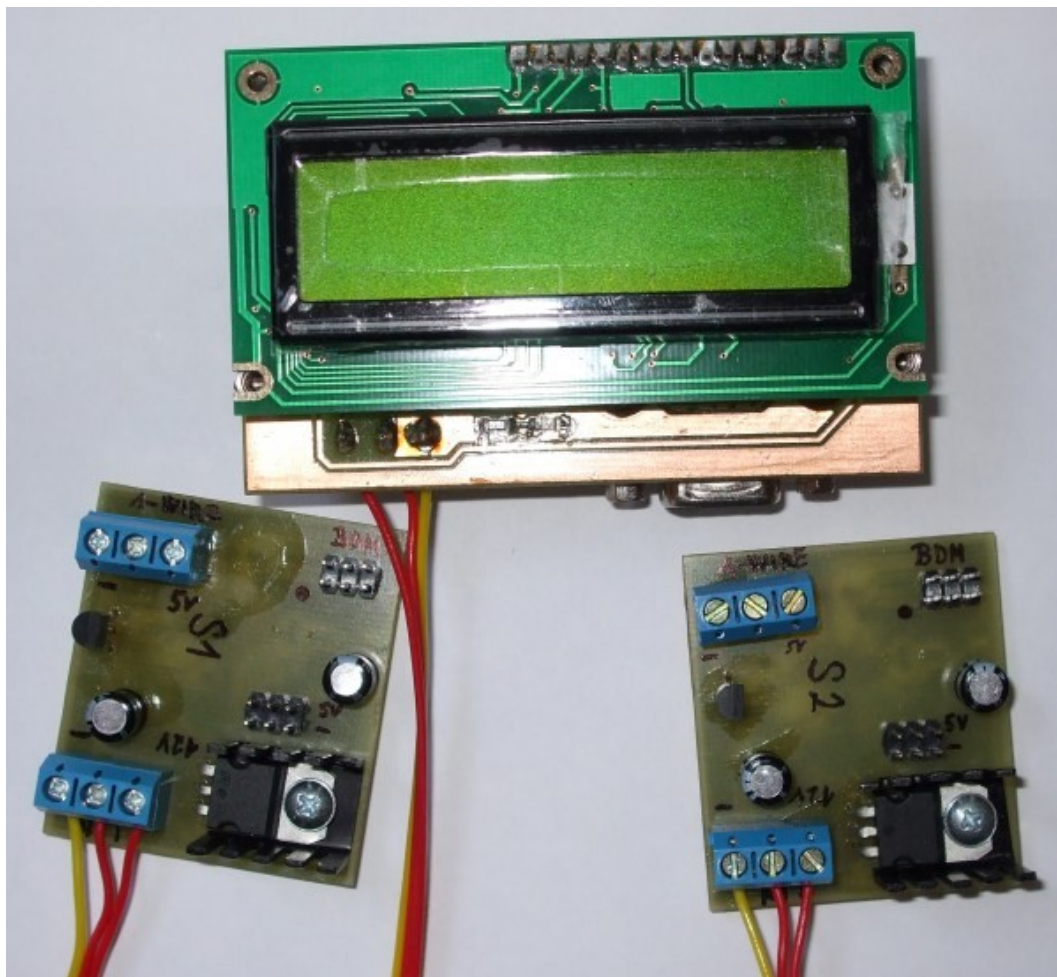


Obr. B.3: Pohled na DPS jednotky slave ze strany SMD



Obr. B.4: Pohled na DPS jednotky slave ze strany součástek

C Příloha



Obr. C.1: Pohled na všechny jednotky

D Příloha

Zde je popsán obsah příloženého CD (obsah jednotlivých adresářů).

Bakalarska_prace – Obsahuje pdf s obsahem této práce.

Datasheets – Obsahuje katalogové listy všech použitých obvodů ve formátu pdf.

Fotky_obrazky – Obsahuje fotky jednotek a obrázky použité v bakalářské práci.

Literatura – Obsahuje dokumenty ve formátu pdf ze kterých jsem čerpal.